

УЧЕБНИК
ДЛЯ ВУЗОВ

 **ПИТЕР®**

СТАНДАРТ ТРЕТЬЕГО ПОКОЛЕНИЯ



Под редакцией С. В. Симоновича

Информатика

Базовый курс

для
БАКАЛАВРОВ
И СПЕЦИАЛИСТОВ

3-е издание

**РЕКОМЕНДОВАНО
МИНИСТЕРСТВОМ ОБРАЗОВАНИЯ И НАУКИ**

ББК 32.973.233я7
УДК 681.3(075)
С37

Симонович С. В.

С37 Информатика. Базовый курс: Учебник для вузов. 3-е изд. Стандарт третьего поколения. — СПб.: Питер, 2011. — 640 с.: ил.

ISBN 978-5-459-00439-7

В учебнике рассмотрены основные категории аппаратных и программных средств вычислительной техники. Указаны базовые принципы построения архитектур вычислительных систем. Обеспечено методическое обоснование процессов взаимодействия информации, данных и методов. Приведены эффективные приемы работы с распространенными программными продуктами. Рассмотрены основные средства, приемы и методы программирования.

Книга предназначена для студентов технических вузов, изучающих информационные технологии в рамках дисциплины «Информатика», для преподавательского состава, для слушателей военных учебных заведений, учреждений системы повышения квалификации и для лиц, изучающих средства вычислительной техники самостоятельно. Третье издание учебника полностью обновлено в соответствии с современной ситуацией в области аппаратных и программных средств, в частности в книге рассмотрена работа в Microsoft Windows 7 и Office 2010.

Рекомендовано Министерством образования и науки Российской Федерации в качестве учебного пособия для студентов высших технических учебных заведений.

ББК 32.973.233я7
УДК 681.3(075)

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-459-00439-7

© ООО Издательство «Питер», 2011

Оглавление

Введение	13
Глава первая. Информация и информатика	16
1.1. Информация в материальном мире	16
Сигналы и данные	16
Данные и методы	16
Понятие об информации	17
Диалектическое единство данных и методов в информационном процессе	19
Свойства информации	20
1.2. Данные	21
Носители данных	21
Операции с данными	22
Кодирование данных двоичным кодом	23
Кодирование целых и действительных чисел	25
Кодирование текстовых данных	25
Универсальная система кодирования текстовых данных	28
Кодирование графических данных	29
Кодирование звуковой информации	30
Основные структуры данных	31
Линейные структуры (списки данных, векторы данных)	32
Табличные структуры (таблицы данных, матрицы данных)	33
Иерархические структуры данных	34
Упорядочение структур данных	35
1.3. Файлы и файловая структура	36
Единицы представления данных	36
Единицы измерения данных	37
Единицы хранения данных	38
Понятие о файловой структуре	38
1.4. Информатика	39
Предмет и задачи информатики	39
Истоки и предпосылки информатики	40
Подведение итогов	41
Вопросы для самоконтроля	42
Глава вторая. Вычислительная техника	43
2.1. История развития средств вычислительной техники	43
Вычислительная система, компьютер	43
Принцип действия компьютера	43
Механические первоисточники	44
Математические первоисточники	46
2.2. Методы классификации компьютеров	47
Классификация по назначению	47
Мини-ЭВМ	49
Микро-ЭВМ	50
Персональные компьютеры	51
Другие виды классификации компьютеров	52
2.3. Состав вычислительной системы	54
Аппаратное обеспечение	54
Программное обеспечение	56
Классификация прикладных программных средств	58
Классификация служебных программных средств	63
Понятие об информационном и математическом обеспечении вычислительных систем	65
Подведение итогов	65
Вопросы для самоконтроля	66

Глава третья. Устройство персонального компьютера	68
3.1. Базовая аппаратная конфигурация персонального компьютера	68
Системный блок	68
Монитор	69
Клавиатура	71
Мышь	75
3.2. Внутренние устройства системного блока	76
Материнская плата	76
Жесткий диск	77
Дисковод гибких дисков	78
Дисководы оптических дисков	80
Видеокарта (видеоадаптер)	82
Звуковая карта	84
3.3. Системы, расположенные на материнской плате	85
Оперативная память	85
Процессор	87
Микросхема ПЗУ и система BIOS	90
Энергонезависимая память CMOS	90
Шинные интерфейсы материнской платы	91
Функции микропроцессорного комплекта (чипсета)	94
3.4. Периферийные устройства персонального компьютера	94
Устройства ввода знаковых данных	94
Устройства командного управления	95
Устройства ввода графических данных	95
Устройства вывода данных	97
Устройства хранения данных	99
Устройства обмена данными	100
Практическое занятие	101
Глава четвертая. Функции операционных систем персональных компьютеров	107
4.1. Обеспечение интерфейса пользователя	107
Режимы работы с компьютером	107
Виды интерфейсов пользователя	108
4.2. Обеспечение автоматического запуска	108
4.3. Организация файловой системы	109
4.4. Обслуживание файловой структуры	110
Создание и именование файлов	110
Создание каталогов (папок)	112
Копирование и перемещение файлов	113
Удаление файлов и каталогов (папок)	113
Навигация по файловой структуре	114
Управление атрибутами файлов	114
4.5. Управление установкой, исполнением и удалением приложений	115
Понятие многозадачности	115
Вопросы надежности	115
Установка приложений	116
Удаление приложений	117
4.6. Взаимодействие с аппаратным обеспечением	117
4.7. Обслуживание компьютера	118
Средства проверки дисков	118
Средства «сжатия» дисков	120
Средства управления виртуальной памятью	120
Средства кэширования дисков	121
Средства резервного копирования данных	121
4.8. Прочие функции операционных систем	121

Подведение итогов	122
Вопросы для самоконтроля	123

Глава пятая. Основы работы с операционной системой Windows 7 124

5.1. Основные объекты и приемы управления Windows	124
Рабочий стол Windows 7	124
Управление Windows 7	125
Значки и ярлыки объектов	126
5.2. Файлы и папки Windows	127
Просмотр папок Windows	127
Окно папки	127
Структура окна	128
5.3. Операции с файловой структурой	130
Система окон Компьютер	130
Приемы работы с файловой системой	131
Приемы повышения эффективности в работе с файловой структурой	134
5.4. Использование Главного меню	137
Структура Главного меню	137
5.5. Установка и удаление приложений Windows	140
Особенности спецификации Windows	140
Стандартные средства установки приложений	141
Удаление приложений Windows	141
5.6. Установка оборудования	142
Средства программной установки оборудования	143
Порядок установки оборудования	143
Практическое занятие	144
Исследовательская работа	149

Глава шестая. Настройка операционной системы Windows 150

6.1. Настройка средств ввода-вывода данных	151
Настройка клавиатуры	151
Настройка мыши	151
Настройка стиля управления операционной системой	151
6.2. Настройка оформления рабочей среды	152
Персонализация настроек Windows	152
Настройка фона Рабочего стола	153
Настройка экранной заставки	154
Тематическое оформление рабочей среды	154
6.3. Настройка элементов управления Windows	158
6.4. Настройка средств автоматизации Windows	162
Автоматический запуск приложений	162
Настройка свойств типов файлов	163
Настройка команды Отправить	164
Автоматизация очистки жесткого диска	164
Запуск приложений по расписанию	166
6.5. Настройка средств поиска данных	167
Механизмы автоматического поиска данных	167
Контекстный поиск данных	167
Сложный контекстный поиск	168
Расширенный поиск	169
Особенности поиска в папках	170
Индексация данных	171
6.6. Настройка шрифтов	171
Растровые и векторные шрифты	172
Типы векторных шрифтов	172
Установка и удаление шрифтов	173

6.7. Прочие настройки Windows	174
Настройка системных часов и системного календаря	174
Настройка национальных стандартов и форматов	175
Практическое занятие	175
Самостоятельная работа	178
Глава седьмая. Стандартные приложения Windows	179
7.1. Стандартные прикладные программы	179
Программа Блокнот	179
Графический редактор Paint	184
Текстовый процессор WordPad	191
7.2. Принципы внедрения и связывания объектов	195
Внедрение объектов	195
Связывание объектов	196
Сравнение методов внедрения и связывания	196
OLE-серверы и OLE-клиенты	197
7.3. Служебные приложения Windows	197
Дефрагментация диска	197
Сведения о системе	198
Таблица символов	199
Восстановление системы	200
Наблюдение за функционированием компьютера и операционной системы	201
Средства командной строки	202
7.4. Стандартные средства мультимедиа	202
Практическое занятие	207
Глава восьмая. Компьютерные сети, Интернет, компьютерная безопасность	214
8.1. Компьютерные сети	214
Назначение компьютерных сетей	214
Основные понятия компьютерных сетей	214
8.2. Сетевые службы, основные понятия	217
Понятие виртуального соединения	217
Модель взаимодействия открытых систем ISO/OSI	218
Особенности виртуальных соединений	219
Сетевые службы	219
8.3. Интернет. Основные понятия	220
Ранняя история Интернета	221
Основы функционирования Интернета	222
Службы Интернета	223
8.4. Технологии проводного подключения к Интернету	229
8.5. Технологии беспроводного подключения к Интернету	232
8.6. Информационная безопасность в Интернете	233
Практическое занятие	242
Глава девятая. Получение информации из Интернета	244
9.1. Основные понятия World Wide Web	244
Веб-страница	244
Гиперссылки	244
Средства просмотра Web	245
9.2. Работа с программой Internet Explorer 8	245
Открытие и просмотр веб-страниц	246
Приемы управления браузером	248
Настройка свойств браузера	249
Прием файлов из Интернета	251
9.3. Поиск информации в World Wide Web	253
Поисковые каталоги	254
Поисковые указатели	255

Рекомендации по приемам эффективного поиска	258
Рекомендации по использованию поисковых систем	259
9.4. Отправка и получение сообщений	259
Работа с программой Почта Windows Live	260
Работа с контактами	264
Практическое занятие	265
Глава десятая. Создание простых текстовых документов	271
10.1. Общие сведения о текстовом процессоре Microsoft Word	271
Рабочее окно процессора Microsoft Word 2010	271
Режимы отображения документов	271
Приемы работы с командами ленты	273
Лента Microsoft Word 2010	273
Основные принципы практической работы с текстовым процессором Microsoft Word	275
Первичная настройка текстового процессора Microsoft Word	277
10.2. Приемы работы с текстами в процессоре Microsoft Word	277
Создание документа	278
Специальные средства ввода текста	279
Специальные средства редактирования текста	283
Средства рецензирования текста	284
Форматирование текста	285
10.3. Приемы и средства автоматизации разработки документов	289
Работа со стилями	289
Шаблоны	292
Темы	293
Практическое занятие	293
Глава одиннадцатая. Создание комплексных текстовых документов	299
11.1. Приемы управления объектами Microsoft Word	299
Особенности объектов Word	299
Взаимодействие объектов Word с текстом и страницей	300
Управление свойствами объектов Microsoft Word	303
Взаимодействие объектов друг с другом	307
11.2. Ввод формул	310
Вставка формул в документ	310
11.3. Работа с таблицами	311
Создание таблиц	313
Редактирование таблиц	314
Форматирование таблиц	315
Ввод и форматирование содержимого таблиц	316
Автоматическое форматирование таблиц	316
11.4. Работа с диаграммами	317
Создание базовой диаграммы	317
Настройка внешнего вида диаграммы	318
11.5. Работа с графическими объектами	319
Работа с фигурами	320
Работа с клипартами	323
Работа с изображениями	325
Практическое занятие	327
Глава двенадцатая. Обработка данных средствами электронных таблиц	334
12.1. Основные понятия электронных таблиц	335
Рабочая книга и рабочий лист. Строки, столбцы, ячейки	335
Ввод, редактирование и форматирование данных	336
12.2. Содержание электронной таблицы	337
Формулы	337
Ссылки на ячейки	337

Абсолютные и относительные ссылки	338
Копирование содержимого ячеек	339
Автоматизация ввода	340
Использование стандартных функций	341
12.3. Печать документов Excel	342
Выбор области печати	343
12.4. Применение электронных таблиц для расчетов	344
Итоговые вычисления	344
Использование надстроек	345
12.5. Построение диаграмм и графиков	346
Выбор типа диаграммы	347
Выбор данных	347
Оформление диаграммы	348
Практическое занятие	349
Глава тринадцатая. Работа с базами данных	358
13.1. Основные понятия баз данных	358
Базы данных и системы управления базами данных	358
Структура простейшей базы данных	359
Свойства полей базы данных	359
Типы данных	360
Безопасность баз данных	362
13.2. Формирование баз данных	363
Режимы работы с базами данных	363
Объекты базы данных	363
Проектирование базы данных	366
Разработка схемы данных	367
13.3. Работа с СУБД Microsoft Access 2010	370
Общие замечания	370
Работа с таблицами	371
Работа с запросами	375
Работа с формами	378
Работа с отчетами	383
Практическое занятие	384
Глава четырнадцатая. Приемы и методы работы со сжатыми данными	392
14.1. Теоретические основы сжатия данных	392
Объекты сжатия	393
Обратимость сжатия	393
Алгоритмы обратимых методов	394
Алгоритм RLE	394
Алгоритм KWE	395
Алгоритм Хаффмана	395
Синтетические алгоритмы	396
14.2. Программные средства сжатия данных	396
Базовые требования к диспетчерам архивов	397
Дополнительные требования к диспетчерам архивов	399
Практическое занятие	399
Исследовательская работа	408
Глава пятнадцатая. Введение в компьютерную графику	412
15.1. Основы представления графических данных	412
Виды компьютерной графики	412
Растровая графика	413
Векторная графика	417

Математические основы векторной графики	417
Фрактальная графика	419
Основные понятия трехмерной графики	419
Программные средства обработки трехмерной графики	425
15.2. Представление графических данных	426
Форматы графических данных	426
Понятие цвета	428
Способы описания цвета	429
Цветовая модель CIE Lab	432
Цветовая модель RGB	432
Цветовая модель HSB	433
Цветовая модель CMYK, цветоделение	434
Цветовая палитра	434
Системы управления цветом	435
Практическое занятие	437
15.3. Средства для работы с растровой графикой	438
Программные средства создания растровых изображений	438
Аппаратные средства получения растровых изображений	439
Программа обработки растровой графики Adobe Photoshop	440
15.4. Средства для работы с векторной графикой	446
Средства создания и обработки векторной графики	446
Основные понятия векторной графики	447
Векторный редактор Adobe Illustrator	448
Практическое занятие	451
Исследовательская работа	455
Практическое занятие	456
Исследовательская работа	460

Глава шестнадцатая. Векторный редактор Corel Draw 464

16.1. Особенности CorelDraw	464
16.2. Элементы управления	465
Панель инструментов	465
Панель свойств	469
Строка меню	470
Специальные средства CorelDraw	470
16.3. Рисование графики	472
Создание и модификация объектов	472
Операции с графикой	473
Копирование, дублирование и клонирование	477
16.4. Заполнение объектов	478
16.5. Операции с текстом	481
16.6. Изменение формы объектов	482
16.7. Операции с группами	485
Пример. Рисование топографической карты	487
Практическое занятие	494

Глава семнадцатая. Автоматизация обработки документов 500

17.1. Преобразование документов в электронную форму	500
Сканирование документов	500
Распознавание документов	503
Работа с программой FineReader	503
Практическое занятие	507
17.2. Автоматизированный перевод документов	510
Работа с программой PROMT 9	511
Контроль качества перевода	513

Упрощенные средства перевода	516
Интеграция системы перевода с другими приложениями	516
Практическое занятие	518
Глава восемнадцатая. Средства автоматизации научно-исследовательских работ	520
18.1. Компьютер как инструмент научной работы	520
18.2. Приемы работы с системой Mathcad	524
Ввод формул	524
Ввод текста	526
Форматирование формул и текста	526
Работа с матрицами	527
Стандартные и пользовательские функции	528
Решение уравнений и систем	528
Построение графиков	529
Аналитические вычисления	531
Практическое занятие	532
Глава девятнадцатая. Публикация веб-документов	548
19.1. Создание веб-документов	548
19.2. Применение языка HTML	550
Структура документа HTML	550
Элементы HTML	550
Функциональные блочные элементы	551
Гипертекстовые ссылки	552
Веб-графика	553
Форматирование текста	556
Списки	558
Таблицы	559
Отображение нескольких документов	560
Интерактивные веб-страницы	561
19.3. Работа в Microsoft Visual Web Developer	563
Создание и редактирование документа	563
Применение шаблонов	566
19.4. Публикация веб-документов	566
Практическое занятие	567
Исследовательская работа	574
Глава двадцатая. Основы программирования	577
20.1. Языки программирования	577
Машинный код процессора	577
Алгоритм и программа	577
Что такое язык программирования	578
Компиляторы и интерпретаторы	578
Уровни языков программирования	580
Поколения языков программирования	580
Обзор языков программирования высокого уровня	581
Языки программирования баз данных	583
Языки программирования для Интернета	584
Языки моделирования	585
Прочие языки программирования	585
Вопросы для самоконтроля	586
20.2. Системы программирования	586
Средства создания программ	586
Интегрированные системы программирования	587

Среды быстрого проектирования	588
Архитектура программных систем	589
Основные системы программирования	590
Вопросы для самоконтроля	591
20.3. Алгоритмическое (модульное) программирование	591
Переменные и константы	591
Числовые данные	592
Арифметические операции	593
Арифметические выражения	593
Логические выражения	594
Строчные выражения	595
Указатели	596
Сложные данные	596
Правила работы со сложными типами	598
Описание переменных	598
Новые типы данных	599
Разделение операторов	600
Блок операторов	600
Область действия переменных	600
Оператор присваивания	601
Комментарии	601
Условный оператор (условные вычисления)	602
Повторяющиеся вычисления (операторы цикла)	604
Первый вид оператора цикла	604
Второй вид оператора цикла	605
Защипливание	605
Исключения	606
Параллельные вычисления	606
Ввод и вывод	607
Вопросы для самоконтроля	607
20.4. Структурное программирование	608
Подпрограммы	608
Нисходящее проектирование	608
Процедуры и функции	610
Параметры подпрограмм	610
Управление последовательностью вызова подпрограмм	610
Структура подпрограммы	611
Как функция возвращает значение	611
Формальные и фактические параметры	612
Событийно-ориентированное программирование	612
Вопросы для самоконтроля	613
20.5. Объектно-ориентированное программирование	613
Понятие объекта	613
Класс	614
Описание нового класса	614
Наследование	615
Полиморфизм	615
Визуальное программирование	615
Вопросы для самоконтроля	616
20.6. Проектирование программ	616
Программирование как вид деятельности	616
Экономические аспекты программирования	617
Этапы разработки программ	617
Период разработки ПО	617
Контроль качества	620
Стандарты качества ПО	621

Повышение индивидуального мастерства	621
Гибкие методики	622
Методы маркетинга программного обеспечения	622
Вопросы для самоконтроля	623
20.7. Пример на Бейсике. Разведение кроликов	623
Постановка задачи	623
Запуск QBasic	624
Вывод на экран	624
Редактор программы	625
Ввод информации от пользователя	625
Главная часть программы	625
Типы данных в Бейсике	626
Добавление новой функции	627
Расчет популяции	627
Сохранение текста программы в файле	628
Запуск программы	628
20.8. Пример на Паскале. Раскрашивание круга	629
Постановка задачи	629
Знакомство с Embarcadero Delphi XE	629
Заголовок окна	630
Размещение компонентов на форме	631
Сохранение проекта	631
Обработка нажатия кнопки	632
Запуск программы	633
20.9. Пример на C++. Рисование графиков	634
Постановка задачи	634
Решение задачи	634
Практические задания по программированию	636

Введение

Коренное отличие информатики от других технических дисциплин, изучаемых в высшей школе, состоит в том, что ее предмет изучения меняется ускоренными темпами. Сегодня количество компьютеров в мире превышает один миллиард единиц, при этом каждая вычислительная система по-своему уникальна. Найти две системы с одинаковыми аппаратными и программными конфигурациями весьма сложно и потому для эффективной эксплуатации вычислительной техники от специалистов требуется достаточно широкий уровень знаний и практических навыков.

Вместе с тем, в количественном отношении темп численного роста вычислительных систем заметно превышает темп подготовки специалистов, способных эффективно работать с ними. При этом в среднем один раз в полтора года удваиваются основные технические параметры аппаратных средств, один раз в два-три года меняются поколения программного обеспечения и один раз в пять-семь лет меняется база стандартов, интерфейсов и протоколов.

Таким образом, кардинальным отличием информатики от других технических дисциплин является тот факт, что ее предметная область изменяется чрезвычайно динамично. Все, кто причастен к преподаванию информатики в высшей школе, хорошо знают, как часто приходится менять содержание учебных планов, рабочих программ, учебно-методической документации. Далеко не всегда удается обеспечить соответствие материально-технической базы учебного процесса текущему состоянию предметной области. И даже своевременное реагирование на научно-технические достижения не всегда позволяет обеспечить уровень знаний и навыков выпускника, адекватный потребностям сферы материального производства и коммерческого рынка, — настолько динамичны процессы в области информационных технологий.

Ныне информатика сталкивается с парадоксальным фактом. Ее основная задача состоит в преодолении общечеловеческого кризисного явления, называемого «информационным бумом», путем внедрения средств и методов, автоматизирующих операции с данными. Однако даже в собственной предметной области информатика испытывает такой информационный бум, какого не знает ни одна другая область человеческой деятельности. Например, мировой ассортимент изданий, имеющих прямое отношение к информатике (не считая периодических и электронных), составляет порядка десяти тысяч томов в год и полностью обновляется не реже, чем раз в два года.

Анализируя вышеуказанные особенности информатики, авторы данного пособия приходят к выводу, что для преподавания информатики в сложившихся условиях необходимо расширенное взаимодействие между учебными программами общетехнических и специальных дисциплин и учебной программой курса информатики. Основные принципы, вытекающие из такого подхода, включают непрерывность и системность образования, а также раннюю профессиональную ориентацию.

Непрерывность образования. Практические приемы работы со средствами вычислительной техники закрепляются не только при изучении информатики, но

и в течение всего периода обучения. Они используются при проведении учебных занятий по самым разным дисциплинам.

Системность образования. В едином методическом подходе, основанном на системе *задача — средство — методы — приемы*, происходит перекрестное взаимодействие изучаемых дисциплин. Конкретная дисциплина поставяет комплекс *задача — методы*, а информатика обеспечивает комплекс *средства — приемы*.

Ранняя профессиональная ориентация. В системе высшего технического образования действует многоуровневая иерархическая система, основанная на том, что знания студента по общетехническим дисциплинам, как правило, реализуются в практические навыки опосредованно, то есть через дисциплины специального цикла, базирующиеся на общетехнических. Информатика — одна из немногих общетехнических дисциплин, развивающая такие практические навыки, которые востребуются напрямую и немедленно, сразу после включения молодого специалиста в профессиональную деятельность.

Свою работу над книгой авторы подчинили реализации указанных принципов. Этому подчинены структура и содержание пособия. В целом книга состоит из двадцати глав, содержащих достаточно полные сведения о современном состоянии аппаратных и программных средств вычислительной техники.

Главы 1, 2, 8, 15 являются теоретическими и обеспечивают единую методическую базу как для изучения информатики, так и для взаимодействия различных учебных дисциплин на платформе информатики.

Главы 9–14, 16, 18 представляют единую технологическую базу для взаимодействия информатики и других предметных дисциплин. Средства, рассмотренные здесь, могут быть использованы при подготовке домашних заданий, контрольных, курсовых и дипломных работ, при обработке результатов экспериментов, сборе исходной информации для самостоятельных исследований, при выполнении графических работ, математическом моделировании физических и технических процессов и при математическом обосновании разрабатываемых проектов.

Главы 3–7, 10, 12, 13, 16, 17, 19, 20 служат тем же задачам, но являются дополнительным средством ранней профессиональной ориентации. Сведения и навыки, полученные в ходе их изучения, могут быть востребованы немедленно после включения выпускника в практическую деятельность. Эти разделы позволяют обеспечить общую уверенность студента в востребованности его знаний по окончании учебного заведения, независимо от обстоятельств и особенностей конкретного трудоустройства.

Главы, имеющие теоретическое и методообразующее содержание, завершаются списком контрольных вопросов, которые могут обсуждаться на лекционных и семинарских занятиях. Главы, имеющие практическое содержание, завершаются упражнениями и исследовательскими работами. Предполагается, что практические упражнения носят инструктивно-методический характер и выполняются под руководством преподавателя (лаборанта), а исследовательские работы имеют творческий характер и комплексное содержание. Они предназначены для самостоятельной работы и предполагают подготовку итогового отчета. Различие между этими видами занятий отражено в балансе отводимого на них времени.

Исходя из структуры и содержания книги, авторы рассчитывают на то, что она будет полезна следующим категориям читателей:

- ☐ студентам технических специальностей вузов, изучающим информатику как самостоятельную дисциплину;
- ☐ преподавательскому составу, осуществляющему теоретическую и практическую подготовку студентов по дисциплине «Информатика»;
- ☐ преподавателям иных дисциплин, использующим персональные компьютеры в качестве технического средства обучения и (или) средства подготовки учебно-методических материалов (бумажных и электронных) по своей предметной области;
- ☐ лицам, самостоятельно изучающим или осваивающим аппаратные и программные средства вычислительной техники.

Глава первая

Информация и информатика

1.1. Информация в материальном мире

Сигналы и данные

Мы живем в материальном мире. Все, что нас окружает и с чем мы сталкиваемся ежедневно, относится либо к *физическим телам*, либо к *физическим полям*. Из курса физики мы знаем, что состояния абсолютного покоя не существует и физические объекты находятся в состоянии непрерывного движения и изменения, которое сопровождается обменом энергией и ее переходом из одной формы в другую.

Все виды *энергообмена* сопровождаются появлением сигналов, то есть все сигналы имеют в своей основе материальную энергетическую природу. При взаимодействии сигналов с физическими телами в последних возникают определенные изменения свойств — это явление называется *регистрацией сигналов*. Такие изменения можно наблюдать, измерять или фиксировать иными способами — при этом возникают и регистрируются новые сигналы, то есть образуются данные.

Данные — это зарегистрированные сигналы.

Данные и методы

Обратим внимание на то, что данные несут в себе *информацию* о событиях, произошедших в материальном мире, поскольку они являются регистрацией сигналов, возникших в результате этих событий. Однако данные не тождественны информации. Наблюдая излучения далеких звезд, человек получает определенный поток данных, но станут ли эти данные информацией, зависит еще от очень многих обстоятельств. Рассмотрим ряд примеров.

Наблюдая за состязаниями бегунов, мы с помощью механического секундомера регистрируем начальное и конечное положение стрелки прибора. В итоге мы замеряем величину ее перемещения за время забега — это регистрация данных. Однако информацию о времени преодоления дистанции мы пока не получаем. Для того чтобы данные о перемещении стрелки дали информацию о времени забега, необходимо наличие *метода* пересчета одной физической величины в другую. Надо знать цену деления шкалы секундомера (или знать метод ее определения) и надо также знать, как умножается цена деления прибора на величину перемещения, то есть надо еще обладать математическим методом умножения.

Если вместо механического секундомера используется электронный, суть дела не меняется. Вместо регистрации перемещения стрелки происходит регистрация количества тактов колебаний, произошедших в электронной системе за время измерения. Даже если секундомер непосредственно отображает время в секундах и нам не нужен метод пересчета, то метод преобразования данных все равно присутствует — он реализован специальными электронными компонентами и работает автоматически, без нашего участия.

Прослушивая передачу радиостанции на незнакомом языке, мы получаем данные, но не получаем информацию в связи с тем, что не владеем методом преобразования данных в известные нам понятия. Если эти данные записать на лист бумаги или на магнитную ленту, изменится форма их представления, произойдет новая регистрация и, соответственно, образуются новые данные. Такое преобразование можно использовать, чтобы все-таки извлечь информацию из данных путем подбора метода, адекватного их новой форме. Для обработки данных, записанных на листе бумаги, адекватным может быть метод перевода со словарем, а для обработки данных, записанных на магнитной ленте, можно пригласить переводчика, обладающего своими методами перевода, основанными на знаниях, полученных в результате обучения или предшествующего опыта.

Если в нашем примере заменить радиопередачу телевизионной трансляцией, ведущейся на незнакомом языке, то мы увидим, что наряду с данными мы все-таки получаем определенную (хотя и не полную) информацию. Это связано с тем, что люди, не имеющие дефектов зрения, априорно владеют адекватным методом восприятия данных, передаваемых электромагнитным сигналом в полосе частот видимого спектра с интенсивностью, превышающей порог чувствительности глаза. В таких случаях говорят, что *метод известен по контексту*, то есть данные, составляющие информацию, имеют свойства, однозначно определяющие адекватный метод получения этой информации. (Для сравнения скажем, что слепому «телезрителю» контекстный метод неизвестен и он оказывается в положении радиослушателя, пример с которым был рассмотрен выше.)

Понятие об информации

Несмотря на то что с понятием информации мы сталкиваемся ежедневно, строгого и общепризнанного ее определения до сих пор не существует, поэтому вместо определения обычно используют *понятие об информации*. Понятия, в отличие от определений, не даются однозначно, а вводятся на примерах, причем каждая научная дисциплина делает это по-своему, выделяя в качестве основных компонентов те, которые наилучшим образом соответствуют ее предмету и задачам. При этом типична ситуация, когда понятие об информации, введенное в рамках одной научной дисциплины, может опровергаться конкретными примерами и фактами, полученными в рамках другой науки. Например, представление об информации как о совокупности данных, повышающих уровень знаний об объективной реальности окружающего мира, характерное для естественных наук, может быть опровергнуто в рамках социальных наук. Нередки также случаи, когда исходные компоненты, составляющие понятие информации, подменяют свойствами информационных объектов, например, когда понятие информации вводят как совокупность данных, которые «могут быть усвоены и преобразованы в знания».

Для информатики как технической науки понятие информации не может основываться на таких антропоцентрических понятиях, как *знание*, и не может опираться только на объективность фактов и свидетельств. Средства вычислительной техники обладают способностью обрабатывать информацию автоматически, без участия человека, и ни о каком знании или незнании здесь речь идти не может. Эти

средства могут работать с искусственной, абстрактной и даже с ложной информацией, не имеющей объективного отражения ни в природе, ни в обществе.

В этой работе мы даем новое определение информации, основанное на ранее продемонстрированном факте взаимодействия данных и методов в момент ее образования.

Информация — это продукт взаимодействия данных и адекватных им методов.

Поскольку в такой форме определение информации дается впервые, читатель приглашается для его всесторонней проверки в рамках других известных ему научных дисциплин, а мы рассмотрим пример, в свое время использованный Норбертом Винером для того, чтобы показать, как информация отдельных членов популяции становится информацией общества.

Допустим, я нахожусь в лесах вдвоем со смышленным дикарем, который не может говорить на моем языке и на языке которого я тоже не могу говорить. Даже без какого-либо условного языка знаков, известного нам обоим, я могу многое узнать от него. Мне нужно лишь быть особо внимательным в те моменты, когда он обнаруживает признаки волнения или интереса. Тогда я должен посмотреть вокруг, особенно в направлении его взгляда, и запомнить все, что я увижу и услышу. Не пройдет много времени, как я открою, какие предметы представляются важными для него, — не потому, что он сообщил мне о них словами, но потому, что я сам их заметил. Иначе говоря, сигнал, лишенный внутреннего содержания, может приобрести для моего спутника смысл по тому, что наблюдает он в данный момент, и может приобрести для меня смысл по тому, что наблюдаю я в данный момент. Способность дикаря замечать моменты моего особенно активного внимания сама по себе образует язык, возможности которого столь же разнообразны, как и диапазон впечатлений, доступных нам обоим.

Н. Винер. Кибернетика

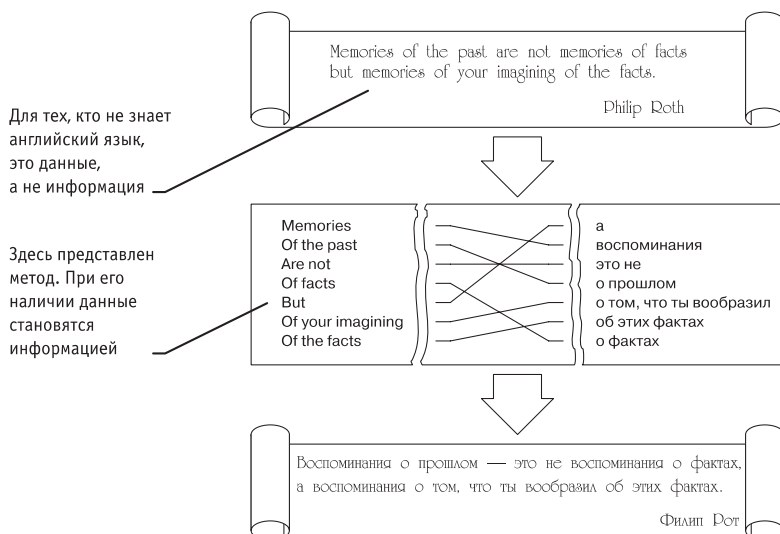


Рис. 1.1. Связь между данными и информацией

Анализируя этот пример, мы видим, что здесь речь идет о данных и методах. Прежде всего, здесь автор прямо говорит о целой группе методов, связанных с наблюдением и анализом, и даже приводит вариант конкретного алгоритма, адекватного рамкам его гипотетического эксперимента (*посмотреть, запомнить, открыть...*). Автор неоднократно подчеркивает требование адекватности метода (дикарь должен быть *смышленным*, а наблюдатель должен быть *особо внимательным*), без которого информация может и не образоваться.

Диалектическое единство данных и методов в информационном процессе

Рассмотрим данное выше определение информации и обратим внимание на следующие обстоятельства.

1. *Динамический характер информации.* Информация не является статичным объектом — она динамически меняется и существует только в момент взаимодействия данных и методов. Все прочее время она пребывает в состоянии данных. Таким образом, информация существует только в момент протекания *информационного процесса*. Все остальное время она содержится в виде данных.
2. *Требование адекватности методов.* Одни и те же данные могут в момент потребления поставлять разную информацию в зависимости от степени адекватности взаимодействующих с ними методов. Например, для человека, не владеющего китайским языком, письмо, полученное из Пекина, дает только ту информацию, которую можно получить методом наблюдения (количество страниц, цвет и сорт бумаги, наличие незнакомых символов и т. п.). Все это информация, но это не вся информация, заключенная в письмо. Использование более адекватных методов даст иную информацию.
3. *Диалектический характер взаимодействия данных и методов.* Обратим внимание на то, что данные являются *объективными*, поскольку это результат регистрации объективно существовавших сигналов, вызванных изменениями в материальных телах или полях. В то же время, методы являются *субъективными*. В основе искусственных методов лежат алгоритмы (упорядоченные последовательности команд), составленные и подготовленные людьми (субъектами). В основе естественных методов лежат биологические свойства субъектов информационного процесса. Таким образом, *информация возникает и существует в момент диалектического взаимодействия объективных данных и субъективных методов*.

Такой дуализм известен своими проявлениями во многих науках. Так, например, в основе важнейшего вопроса философии о первичности материалистического и идеалистического подходов к теории познания лежит не что иное, как двойственный характер информационного процесса. В обоснованиях обоих подходов нетрудно обнаружить упор либо на объективность данных, либо на субъективность методов. Подход к информации как к объекту особой природы, возникающему в результате диалектического взаимодействия объективных данных с субъективными методами, позволяет во многих случаях снять противоречия, возникающие в философских обоснованиях ряда научных теорий и гипотез.

Свойства информации

Итак, информация является динамическим объектом, образующимся в момент взаимодействия объективных данных и субъективных методов. Как и всякий объект, она обладает свойствами (объекты различимы по своим свойствам). Характерной особенностью информации, отличающей ее от других объектов природы и общества, является отмеченный выше дуализм: на свойства информации влияют как свойства данных, составляющих ее содержательную часть, так и свойства методов, взаимодействующих с данными в ходе информационного процесса. По окончании процесса свойства информации переносятся на свойства новых данных, то есть свойства методов могут переходить на свойства данных.

Можно привести немало разнообразных свойств информации. Каждая научная дисциплина рассматривает те свойства, которые ей наиболее важны. С точки зрения информатики наиболее важными представляются следующие свойства: объективность, полнота, достоверность, адекватность, доступность и актуальность информации.

Объективность и субъективность информации. Понятие объективности информации является относительным. Это понятно, если учесть, что методы являются субъективными. Более объективной принято считать ту информацию, в которую методы вносят меньший субъективный элемент. Так, например, принято считать, что в результате наблюдения фотоснимка природного объекта или явления образуется более объективная информация, чем в результате наблюдения рисунка того же объекта, выполненного человеком. В ходе информационного процесса степень объективности информации всегда понижается. Это свойство учитывают, например, в правовых дисциплинах, где по-разному обрабатываются показания лиц, непосредственно наблюдавших события или получивших информацию косвенным путем (посредством умозаключений или со слов третьих лиц). В не меньшей степени объективность информации учитывают в исторических дисциплинах. Одни и те же события, зафиксированные в исторических документах разных стран и народов, выглядят совершенно по-разному. У историков имеются свои методы для тестирования объективности исторических данных и создания новых, более достоверных данных путем сопоставления, фильтрации и селекции исходных данных. Обратим внимание на то, что здесь речь идет не о повышении объективности данных, а о повышении их достоверности (это совсем другое свойство).

Полнота информации. Полнота информации во многом характеризует *качество информации* и определяет *достаточность* данных для принятия решений или для создания новых данных на основе имеющихся. Чем полнее данные, тем шире диапазон методов, которые можно использовать, тем проще подобрать метод, вносящий минимум погрешностей в ход информационного процесса.

Достоверность информации. Данные возникают в момент регистрации сигналов, но не все сигналы являются «полезными» — всегда присутствует какой-то уровень посторонних сигналов, в результате чего полезные данные сопровождаются определенным уровнем «информационного шума». Если полезный сигнал зарегистрирован более четко, чем посторонние сигналы, достоверность информации может быть более высокой. При увеличении уровня шумов достоверность информации

снижается. В этом случае для передачи того же количества информации требуется использовать либо больше данных, либо более сложные методы.

Адекватность информации — это степень соответствия реальному объективному состоянию дела. Неадекватная информация может образовываться при создании новой информации на основе неполных или недостоверных данных. Однако и полные, и достоверные данные могут приводить к созданию неадекватной информации в случае применения к ним неадекватных методов.

Доступность информации — мера возможности получить ту или иную информацию. На степень доступности информации влияют одновременно как доступность данных, так и доступность адекватных методов для их интерпретации. Отсутствие доступа к данным или отсутствие адекватных методов обработки данных приводит к одинаковому результату: информация оказывается недоступной. Отсутствие адекватных методов для работы с данными во многих случаях приводит к применению неадекватных методов, в результате чего образуется неполная, неадекватная или недостоверная информация.

Актуальность информации — это степень соответствия информации текущему моменту времени. Нередко с актуальностью, как и с полнотой, связывают коммерческую ценность информации. Поскольку информационные процессы растянуты во времени, то достоверная и адекватная, но устаревшая информация может приводить к ошибочным решениям. Необходимость поиска (или разработки) адекватного метода для работы с данными может приводить к такой задержке в получении информации, что она становится неактуальной и ненужной. На этом, в частности, основаны многие современные системы шифрования данных с *открытым ключом*. Лица, не владеющие ключом (методом) для чтения данных, могут заняться поиском ключа, поскольку алгоритм его работы доступен, но продолжительность этого поиска столь велика, что за время работы информация теряет актуальность и, соответственно, связанную с ней практическую ценность.

1.2. Данные

Носители данных

Данные — диалектическая составная часть информации. Они представляют собой зарегистрированные сигналы. При этом физический метод регистрации может быть любым: механическое перемещение физических тел, изменение их формы или параметров качества поверхности, изменение электрических, магнитных, оптических характеристик, химического состава и (или) характера химических связей, изменение состояния электронной системы и многое другое. В соответствии с методом регистрации данные могут храниться и транспортироваться на носителях различных видов.

Самым распространенным носителем данных, хотя и не самым экономичным, по-видимому, является бумага. На бумаге данные регистрируются путем изменения оптических характеристик ее поверхности. Изменение оптических свойств (изменение коэффициента отражения поверхности в определенном диапазоне длин волн) используется также в устройствах, осуществляющих запись лазерным лучом

на пластмассовых носителях с отражающим покрытием (*CD-ROM, DVD-ROM*). В качестве носителей, использующих изменение магнитных свойств, можно назвать магнитные ленты и диски. Регистрация данных путем изменения химического состава поверхностных веществ носителя широко используется в фотографии. На биохимическом уровне происходит накопление и передача данных в живой природе.

Носители данных интересуют нас не сами по себе, а постольку, поскольку свойства информации весьма тесно связаны со свойствами ее носителей. Любой носитель можно характеризовать параметром *разрешающей способности* (количеством данных, записанных в принятой для носителя единице измерения) и *динамическим диапазоном* (логарифмическим отношением интенсивности амплитуд максимального и минимального регистрируемого сигналов). От этих свойств носителя нередко зависят такие свойства информации, как полнота, доступность и достоверность. Например, мы можем рассчитывать на то, что в базе данных, размещаемой на компакт-диске, проще обеспечить полноту информации, чем в аналогичной по назначению базе данных, размещенной на гибком магнитном диске, поскольку в первом случае плотность записи данных на единице длины дорожки намного выше. Для обычного потребителя доступность информации в книге заметно выше, чем той же информации на компакт-диске, поскольку не все потребители обладают необходимым оборудованием. И наконец, известно, что визуальный эффект от просмотра слайда в проекторе намного больше, чем от просмотра аналогичной иллюстрации, напечатанной на бумаге, поскольку диапазон яркостных сигналов в проходящем свете на два-три порядка больше, чем в отраженном.

Задача преобразования данных с целью смены носителя относится к одной из важнейших задач информатики. В структуре стоимости вычислительных систем устройства для ввода и вывода данных, работающие с носителями информации, составляют до половины стоимости аппаратных средств.

Операции с данными

В ходе информационного процесса данные преобразуются из одного вида в другой с помощью методов. Обработка данных включает в себя множество различных операций. По мере развития научно-технического прогресса и общего усложнения связей в человеческом обществе трудозатраты на обработку данных неуклонно возрастают. Прежде всего, это связано с постоянным усложнением условий управления производством и обществом. Второй фактор, также вызывающий общее увеличение объемов обрабатываемых данных, тоже связан с научно-техническим прогрессом, а именно с быстрыми темпами появления и внедрения новых носителей данных, средств их хранения и доставки.

В структуре возможных операций с данными можно выделить следующие основные:

- ❑ *сбор данных* — накопление информации с целью обеспечения достаточной полноты для принятия решений;
- ❑ *формализация данных* — приведение данных, поступающих из разных источников, к одинаковой форме, чтобы сделать их сопоставимыми между собой, то есть повысить их уровень доступности;

- ❑ *фильтрация данных* — отсеивание «лишних» данных, в которых нет необходимости для принятия решений; при этом должен уменьшаться уровень «шума», а достоверность и адекватность данных должны возрастать;
- ❑ *сортировка данных* — упорядочение данных по заданному признаку с целью удобства использования; повышает доступность информации;
- ❑ *архивация данных* — организация хранения данных в удобной и легкодоступной форме; служит для снижения экономических затрат по хранению данных и повышает общую надежность информационного процесса в целом;
- ❑ *защита данных* — комплекс мер, направленных на предотвращение утраты, воспроизведения и модификации данных;
- ❑ *транспортировка данных* — прием и передача (доставка и поставка) данных между удаленными участниками информационного процесса; при этом источник данных в информатике принято называть *сервером*, а потребителя — *клиентом*;
- ❑ *преобразование данных* — перевод данных из одной формы в другую или из одной структуры в другую. Преобразование данных часто связано с изменением типа носителя: например книги можно хранить в обычной бумажной форме, но можно использовать для этого и электронную форму, и микрофото пленку. Необходимость в многократном преобразовании данных возникает также при их транспортировке, особенно если она осуществляется средствами, не предназначенными для транспортировки данного вида данных. В качестве примера можно упомянуть, что для транспортировки цифровых потоков данных по каналам телефонных сетей (которые изначально были ориентированы только на передачу аналоговых сигналов в узком диапазоне частот) необходимо преобразование цифровых данных в некое подобие звуковых сигналов, чем и занимаются специальные устройства — *телефонные модемы*.

Приведенный здесь список типовых операций с данными далеко не полон. Миллионы людей во всем мире занимаются созданием, обработкой, преобразованием и транспортировкой данных, и на каждом рабочем месте выполняются свои специфические операции, необходимые для управления социальными, экономическими, промышленными, научными и культурными процессами. Полный список возможных операций составить невозможно, да и не нужно. Сейчас нам важен другой вывод: *работа с информацией может иметь огромную трудоемкость и ее надо автоматизировать.*

Кодирование данных двоичным кодом

Для автоматизации работы с данными, относящимися к различным типам, очень важно унифицировать их форму представления — для этого обычно используется прием *кодирования*, то есть выражение данных одного типа через данные другого типа. Естественные человеческие *языки* — это не что иное, как системы кодирования понятий для выражения мыслей посредством речи. К языкам близко примыкают *азбуки* (системы кодирования компонентов языка с помощью графических символов). История знает интересные, хотя и безуспешные попытки создания «универсальных» языков и азбук. По-видимому, безуспешность попыток их вне-

дрения связана с тем, что национальные и социальные образования естественным образом понимают, что изменение системы кодирования общественных данных непременно приводит к изменению общественных методов (то есть норм права и морали), а это может быть связано с социальными потрясениями.

Та же проблема универсального средства кодирования достаточно успешно реализуется в отдельных отраслях техники, науки и культуры. В качестве примеров можно привести систему записи математических выражений, телеграфную азбуку, морскую флажковую азбуку, систему Брайля для слепых и многое другое.

Своя система существует и в вычислительной технике — она называется *двоичным кодированием* и основана на представлении данных последовательностью всего двух знаков: 0 и 1. Эти знаки называются *двоичными цифрами*, по английски — *binary digit* или, сокращенно, *bit* (*бум*).

С	О	М	Р	У	Т	Е	Р	
43	4F	4D	50	55	54	45	52	Код ASCII
— . . .	— — —	— —	. — — .	. . —	—	.	. — .	Код Морзе
• •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	Код Брайля
								Код морской сигнальный

Рис. 1.2. Примеры различных систем кодирования

Одним битом могут быть выражены два понятия: 0 или 1 (*да* или *нет*, *черное* или *белое*, *истина* или *ложь* и т. п.). Если количество битов увеличить до двух, то уже можно выразить четыре различных понятия:

00 01 10 11

Тремя битами можно закодировать восемь различных значений:

000 001 010 011 100 101 110 111

Увеличивая на единицу количество разрядов в системе двоичного кодирования, мы увеличиваем в два раза количество значений, которое может быть выражено в данной системе, то есть общая формула имеет вид:

$$N = 2^m,$$

где N — количество независимых кодируемых значений; m — разрядность двоичного кодирования, принятая в данной системе.

Кодирование целых и действительных чисел

Целые числа кодируются двоичным кодом достаточно просто — достаточно взять целое число и делить его пополам до тех пор, пока в остатке не образуется ноль или единица. Совокупность остатков от каждого деления, записанная справа налево вместе с последним остатком, и образует двоичный аналог десятичного числа.

$$19 : 2 = 9 + 1$$

$$9 : 2 = 4 + 1$$

$$4 : 2 = 2 + 0$$

$$2 : 2 = 1$$

Таким образом, $19_{10} = 1011_2$.

Для кодирования целых чисел от 0 до 255 достаточно иметь 8 разрядов двоичного кода (8 бит). Шестнадцать бит позволяют закодировать целые числа от 0 до 65 535, а 24 бита — уже более 16,5 млн разных значений.

Для кодирования действительных чисел используют 80-разрядное кодирование. При этом число предварительно преобразуется в *нормализованную форму*:

$$3,1415926 = 0,31415926 \cdot 10^1$$

$$300\,000 = 0,3 \cdot 10^6$$

$$123\,456\,789 = 0,123456789 \cdot 10^{10}$$

Первая часть числа называется *мантиссой*, а вторая — *характеристикой*. Большую часть из 80 бит отводят для хранения мантиссы (вместе со знаком) и некоторое фиксированное количество разрядов отводят для хранения характеристики (тоже со знаком).

Кодирование текстовых данных

Если каждому символу алфавита сопоставить определенное целое число (например, порядковый номер), то с помощью двоичного кода можно кодировать и текстовую информацию. Восьми двоичных разрядов достаточно для кодирования 256 различных символов. Этого хватит, чтобы выразить различными комбинациями восьми битов все символы английского и русского языков, как строчные, так и прописные, а также знаки препинания, символы основных арифметических действий и некоторые общепринятые специальные символы, например символ «§».

Технически это выглядит очень просто, однако всегда существовали достаточно веские организационные сложности. В первые годы развития вычислительной техники они были связаны с отсутствием необходимых стандартов, а в настоящее время вызваны, наоборот, избытком одновременно действующих и противоречивых стандартов. Для того чтобы весь мир одинаково кодировал текстовые данные, нужны единые таблицы кодирования, а это пока невозможно из-за противоречий между символами национальных алфавитов, а также противоречий корпоративного характера.

Для английского языка, захватившего де-факто нишу международного средства общения, противоречия уже сняты. Институт стандартизации США (*ANSI — American National Standard Institute*) ввел в действие систему кодирования *ASCII* (*American Standard Code for Information Interchange* — стандартный код информационного обмена США). В системе ASCII закреплены две таблицы кодирования — *базовая* и *расширенная*. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 до 255.

Первые 32 кода базовой таблицы, начиная с нулевого, отданы производителям аппаратных средств (в первую очередь производителям компьютеров и печатающих устройств). В этой области размещаются так называемые *управляющие коды*, которым не соответствуют никакие символы языков, и, соответственно, эти коды не выводятся ни на экран, ни на устройства печати, но ими можно управлять тем, как производится вывод прочих данных.

Начиная с кода 32 по код 127 размещены коды символов английского алфавита, знаков препинания, цифр, арифметических действий и некоторых вспомогательных символов. Базовая таблица кодировки ASCII приведена в табл. 1.1.

Таблица 1.1. Базовая таблица кодировки ASCII

32	пробел	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	

Аналогичные системы кодирования текстовых данных были разработаны и в других странах. Так, например, в СССР в этой области действовала система кодирования КОИ-7 (*код обмена информацией, семизначный*). Однако поддержка производителей оборудования и программ вывела американский код *ASCII* на уровень международного стандарта, и национальным системам кодирования пришлось «отступить» во вторую, расширенную часть системы кодирования, определяющую значения кодов со 128 по 255. Отсутствие единого стандарта в этой области привело к множественности одновременно действующих кодировок. Только в России можно указать три действующих стандарта кодировки и еще два устаревших.

Так, например, кодировка символов русского языка, известная как кодировка *Windows-1251*, была введена «извне» — компанией *Microsoft*, но, учитывая широкое распространение операционных систем и других продуктов этой компании в России, она глубоко закрепились и нашла широкое распространение (табл. 1.2). Эта

кодировка используется на большинстве локальных компьютеров, работающих на платформе *Windows*.

Таблица 1.2. Кодировка Windows 1251

128 Ъ	144 ђ	160 °	176 А	192 Р	224 а	240 р
129 Ѓ	145 ‘	161 ў	177 ±	193 Б	225 б	241 с
130 ,	146 ’	162 ъ	178 І	194 В	226 в	242 т
131 ҃	147 “	163 Ј	179 і	195 Г	227 г	243 у
132 „	148 ”	164 ѝ	180 ґ	196 Д	228 д	244 ф
133 ...	149 •	165 ґ	181 μ	197 Е	229 е	245 х
134 †	150 —	166 —	182 ¶	198 Ж	230 ж	246 ц
135 ‡	151 —	167 §	183 ·	199 З	231 з	247 ч
136 ‘	152 ‘	168 Ё	184 ё	200 И	232 и	248 ш
137 ‰	153 ™	169 ©	185 №	201 Й	233 й	249 щ
138 Љ	154 ъ	170 €	186 €	202 К	234 к	250 ь
139 ‹	155 ‹	171 «	187 »	203 Л	235 л	251 ы
140 Њ	156 ъ	172 ¬	188 ј	204 М	236 м	252 ь
141 К	157 к	173 -	189 S	205 Н	237 н	253 э
142 Ң	158 ң	174 ®	190 s	206 О	238 о	254 ю
143 Ы	159 ы	175 і	191 і	207 П	239 п	255 я

Другая распространенная кодировка носит название КОИ-8 (*код обмена информацией, восьмизначный*) — ее происхождение относится ко временам действия Совета Экономической Взаимопомощи государств Восточной Европы (табл. 1.3). Сегодня кодировка КОИ-8 имеет широкое распространение в компьютерных сетях на территории России и в российском секторе Интернета.

Международный стандарт, в котором предусмотрена кодировка символов русского алфавита, носит название кодировки *ISO (International Standard Organization — Международный институт стандартизации)*. На практике данная кодировка используется редко (табл. 1.4).

На устаревших компьютерах, работающих в операционных системах MS-DOS, могут действовать еще две кодировки (кодировка ГОСТ и кодировка ГОСТ-альтернативная). Первая из них считалась устаревшей даже в первые годы появления персональной вычислительной техники, но вторая используется и по сей день (табл. 1.5).

Таблица 1.3. Кодировка КОИ-8

128	144	160	176	192	208	224	240
129	145	161	177	193	209	225	241
130	146	162	178	194	210	226	242
131	147	163	179	195	211	227	243
132	148	164	180	196	212	228	244
133	149	165	181	197	213	229	245
134	150	166	182	198	214	230	246
135	151	167	183	199	215	231	247
136	152	168	184	200	216	232	248
137	153	169	185	201	217	233	249
138	154	170	186	202	218	234	250
139	155	171	187	203	219	235	251
140	156	172	188	204	220	236	252
141	157	173	189	205	221	237	253
142	158	174	190	206	222	238	254
143	159	175	191	207	223	239	255

Таблица 1.4. Кодировка ISO

В ISO не определены	160	176	192	208	224	240
	161 Ě	177 А	193 Р	209 а	225 р	241 №
	162 Ъ	178 Б	194 С	210 б	226 с	242 ё
	163 Ѓ	179 В	195 Т	211 в	227 т	243 ђ
	164 Ё	179 Г	195 У	211 г	227 у	243 ř
	165 S	180 Д	196 Ф	212 д	228 ф	244 є
	166 I	181 Е	197 Х	213 е	229 х	245 s
	167 Ĭ	182 Ж	198 Ц	214 ж	230 ц	246 i
	168 J	183 З	199 Ч	215 з	231 ч	247 ĭ
	169 Љ	184 И	200 Ш	216 и	232 ш	248 j
	170 Њ	185 Й	201 Щ	217 ъ	233 щ	249 љ
	171 Ћ	186 К	202 Ъ	218 к	234 њ	250 њ
	172 Ĳ	187 Л	203 Ы	219 л	235 ы	251 ħ
	173 -	188 М	204 Ь	220 м	236 ь	252 k
	174 Ÿ	189 Н	205 Э	221 н	237 э	253 §
	175 Ɔ	190 О	206 Ю	222 о	238 ю	254 Ÿ
		191 П	207 Я	223 п	239 я	255 Ɔ

Таблица 1.5. Гост-альтернативная кодировка

128 А	144 Р	160 а	176 ☐	192 Ь	208 ⊥	224 р	240 Ě
129 Б	145 С	161 б	177 ☐	193 Г	209 ⊥	225 с	241 ё
130 В	146 Т	162 в	178 ☐	194 Т	210 ⊥	226 т	242 Ё
131 Г	147 У	163 г	179	195	211 ⊥	227 у	243 е
132 Д	148 Ф	164 д	180 ⊥	196 —	212 ⊥	228 ф	244 Ĭ
133 Е	149 Х	165 е	181 ⊥	197 +	213 ⊥	229 х	245 i
134 Ж	150 Ц	166 ж	182 ⊥	198 ⊥	214 ⊥	230 ц	246 Ÿ
135 З	151 Ч	167 з	183 ⊥	199 ⊥	215 ⊥	231 ч	247 Ÿ
136 И	152 Ш	168 и	184 ⊥	200 ⊥	216 ⊥	232 ш	248 °
137 Й	153 Щ	169 й	185 ⊥	201 ⊥	217 ⊥	233 щ	249 •
138 К	154 Ъ	170 к	186	202 ⊥	218 ⊥	234 њ	250 •
139 Л	155 Ы	171 л	187 ⊥	203 ⊥	219 ⊥	235 ы	251 √
140 М	156 Ь	172 м	188 ⊥	204 ⊥	220 ⊥	236 ь	252 №
141 Н	157 Э	173 н	189 ⊥	205 —	221 ⊥	237 э	253 ☐
142 О	158 Ю	174 о	190 ⊥	206 ⊥	222 ⊥	238 ю	254 ■
143 П	159 Я	175 п	191 ⊥	207 ⊥	223 ⊥	239 я	255

В связи с избытием систем кодирования текстовых данных, действующих в России, возникает задача межсистемного преобразования данных — это одна из распространенных задач информатики.

Универсальная система кодирования текстовых данных

Если проанализировать организационные трудности, связанные с созданием единой системы кодирования текстовых данных, то можно прийти к выводу, что они вызваны ограниченным набором кодов (256). В то же время очевидно, что если, например, кодировать символы не восьмиразрядными двоичными числами, а числами с большим количеством разрядов, то и диапазон возможных значений кодов станет намного больше. Такая система, основанная на 16-разрядном кодировании символов, получила название *универсальной* — *UNICODE*. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65536 различных символов — этого поля достаточно для размещения в одной таблице символов большинства языков планеты.

Несмотря на тривиальную очевидность такого подхода, простой механический переход на данную систему долгое время сдерживался из-за недостаточных

ресурсов средств вычислительной техники (в системе кодирования UNICODE все текстовые документы автоматически становятся вдвое длиннее). Во второй половине 90-х годов технические средства достигли необходимого уровня обеспеченности ресурсами, и сегодня мы наблюдаем постепенный перевод документов и программных средств на универсальную систему кодирования. О широкой распространенности UNICODE говорит и тот факт, что более половины веб-сайтов в 2010 году используют эту систему кодирования.

Для индивидуальных пользователей это еще больше добавило забот по согласованию документов, выполненных в разных системах кодирования, с программными средствами, но это надо понимать как трудности переходного периода.

Кодирование графических данных

Если рассмотреть с помощью увеличительного стекла черно-белое графическое изображение, напечатанное в газете или книге, то можно увидеть, что оно состоит из мельчайших точек, образующих характерный узор, называемый *растром* (рис. 1.3).

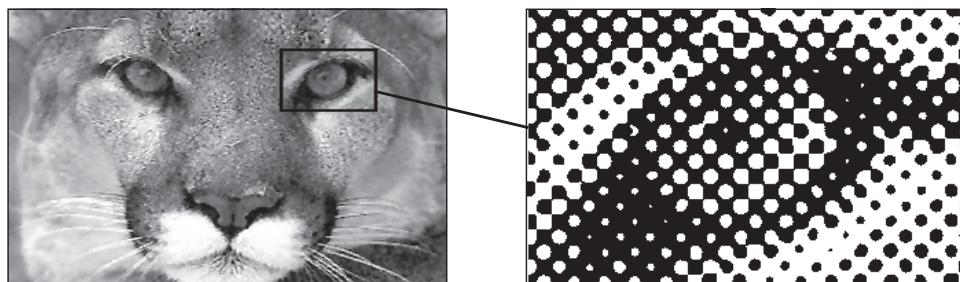


Рис. 1.3. Растр — это метод кодирования графической информации, издавна принятый в полиграфии

Поскольку линейные координаты и индивидуальные свойства каждой точки (яркость) можно выразить с помощью целых чисел, то можно сказать, что растровое кодирование позволяет использовать двоичный код для представления графических данных. Общепринятым на сегодняшний день считается представление черно-белых иллюстраций в виде комбинации точек с 256 градациями серого цвета, и, таким образом, для кодирования яркости любой точки обычно достаточно восьмизначного двоичного числа.

Для кодирования цветных графических изображений применяется *принцип декомпозиции* произвольного цвета на основные составляющие. В качестве таких составляющих используют три основных цвета: красный (*Red, R*), зеленый (*Green, G*) и синий (*Blue, B*). На практике считается (хотя теоретически это не совсем так), что любой цвет, видимый человеческим глазом, можно получить путем механического смешения этих трех основных цветов. Такая система кодирования называется системой *RGB* по первым буквам названий основных цветов.

Если для кодирования яркости каждой из основных составляющих использовать по 256 значений (восемь двоичных разрядов), как это принято для полу-

тоновых черно-белых изображений, то на кодирование цвета одной точки надо затратить 24 разряда. При этом система кодирования обеспечивает однозначное определение 16,5 млн различных цветов, что на самом деле близко к чувствительности человеческого глаза. Режим представления цветной графики с использованием 24 двоичных разрядов называется *полноцветным (True Color)*.

Каждому из основных цветов можно поставить в соответствие дополнительный цвет, то есть цвет, дополняющий основной цвет до белого. Нетрудно заметить, что для любого из основных цветов дополнительным будет цвет, образованный суммой пары остальных основных цветов. Соответственно, дополнительными цветами являются: голубой (*Cyan, C*), пурпурный (*Magenta, M*) и желтый (*Yellow, Y*). Принцип декомпозиции произвольного цвета на составляющие компоненты можно применять не только для основных цветов, но и для дополнительных, то есть любой цвет можно представить в виде суммы голубой, пурпурной и желтой составляющей. Такой метод кодирования цвета принят в полиграфии, но в полиграфии используется еще и четвертая краска — черная (*Black, K*). Поэтому данная система кодирования обозначается четырьмя буквами *СМУК* (черный цвет обозначается буквой *K*, потому что буква *B* уже занята синим цветом), и для представления цветной графики в этой системе надо иметь 32 двоичных разряда. Такой режим тоже называется *полноцветным (True Color)*.

Если уменьшить количество двоичных разрядов, используемых для кодирования цвета каждой точки, то можно сократить объем данных, но при этом диапазон кодируемых цветов заметно сокращается. Кодирование цветной графики 16-разрядными двоичными числами называется режимом *High Color*.

При кодировании информации о цвете с помощью восьми бит данных можно передать только 256 цветовых оттенков. Такой метод кодирования цвета называется *индексным*. Смысл названия в том, что, поскольку 256 значений совершенно недостаточно, чтобы передать весь диапазон цветов, доступный человеческому глазу, код каждой точки раstra выражает не цвет сам по себе, а только его номер (*индекс*) в некоей справочной таблице, называемой *палитрой*. Разумеется, эта палитра должна прикладываться к графическим данным — без нее нельзя воспользоваться методами воспроизведения информации на экране или бумаге (то есть воспользоваться, конечно, можно, но из-за неполноты данных полученная информация не будет адекватной: листва на деревьях может оказаться красной, а небо — зеленым).

Кодирование звуковой информации

Приемы и методы работы со звуковой информацией пришли в вычислительную технику наиболее поздно. К тому же, в отличие от числовых, текстовых и графических данных, у звукозаписей не было столь же длительной и проверенной истории кодирования. В итоге методы кодирования звуковой информации двоичным кодом далеки от стандартизации. Множество отдельных компаний разработали свои корпоративные стандарты, но если говорить обобщенно, то можно выделить два основных направления.

Метод FM (*Frequency Modulation*) основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических

сигналов разных частот, каждый из которых представляет собой правильную синусоиду, а следовательно, может быть описан числовыми параметрами, то есть кодом. В природе звуковые сигналы имеют непрерывный спектр, то есть являются аналоговыми. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства — *аналогово-цифровые преобразователи (АЦП)*. Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют *цифро-аналоговые преобразователи (ЦАП)*. При таких преобразованиях неизбежны потери информации, связанные с методом кодирования. Данный метод кодирования обеспечивает достаточно компактный код, и потому он нашел применение еще в те годы, когда ресурсы средств вычислительной техники были явно недостаточны. В наши дни качество цифрового звука находится на весьма высоком уровне, что открыло возможности для его повсеместного использования.

Метод таблично-волнового (*Wave-Table*) синтеза использует образцы звуков множества различных инструментов, которые хранятся в заранее подготовленных таблицах. В технике такие образцы называют *сэмплами*. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звука. Поскольку в качестве образцов используются «реальные» звуки, то качество звука, полученного в результате синтеза, получается очень высоким и приближается к качеству звучания реальных музыкальных инструментов.

Основные структуры данных

Работа с большими наборами данных автоматизируется проще, когда данные *упорядочены*, то есть образуют заданную структуру. Существует три основных типа структур данных: *линейная*, *иерархическая* и *табличная*. Их можно рассмотреть на примере обычной книги.

Если разобрать книгу на отдельные листы и перемешать их, книга потеряет свое назначение. Она по-прежнему будет представлять набор данных, но подобрать адекватный метод для получения из нее информации весьма непросто. (Еще хуже дело будет обстоять, если из книги вырезать каждую букву отдельно, — в этом случае вряд ли вообще найдется адекватный метод для ее прочтения.)

Если же собрать все листы книги в правильной последовательности, мы получим простейшую структуру данных — *линейную*. Такую книгу уже можно читать, хотя для поиска нужных данных ее придется прочитать подряд, начиная с самого начала, что не всегда удобно.

Для быстрого поиска данных существует *иерархическая структура*. Так, например, книги разбивают на части, разделы, главы, параграфы и т. п. Элементы структуры более низкого уровня входят в элементы структуры более высокого уровня: разделы состоят из глав, главы из параграфов и т. д.

Для больших массивов поиск данных в иерархической структуре намного проще, чем в линейной, однако и здесь необходима *навигация*, связанная с необходимостью просмотра. На практике задачу упрощают тем, что в большинстве книг есть

вспомогательная перекрестная *таблица*, связывающая элементы иерархической структуры с элементами линейной структуры, то есть связывающая разделы, главы и параграфы с номерами страниц. В книгах с простой иерархической структурой, рассчитанных на последовательное чтение, эту таблицу принято называть *оглавлением*, а в книгах со сложной структурой, допускающей выборочное чтение, ее называют *содержанием*.

Линейные структуры (списки данных, векторы данных)

Линейные структуры — это хорошо знакомые нам списки. *Список* — это простейшая структура данных, отличающаяся тем, что каждый элемент данных однозначно определяется своим номером в массиве. Проставляя номера на отдельных страницах рассыпанной книги, мы создаем структуру списка. Обычный журнал посещаемости занятий, например, имеет структуру списка, поскольку все студенты группы зарегистрированы в нем под своими *уникальными* номерами. Мы называем номера *уникальными* потому, что в одной группе не могут быть зарегистрированы два студента с одним и тем же номером.

При создании любой структуры данных надо решить два вопроса: как разделять элементы данных между собой и как разыскивать нужные элементы. В журнале посещаемости, например, это решается так: каждый новый элемент списка заносится с новой строки, то есть разделителем является конец строки. Тогда нужный элемент можно разыскать по номеру строки.

№ п/п	Фамилия, Имя, Отчество
1	Аистов Александр Алексеевич
2	Бобров Борис Борисович
3	Воробьева Валентина Владиславовна
...	
27	Сорокин Сергей Семенович

Разделителем может быть и какой-нибудь специальный символ. Нам хорошо известны разделители между словами — это пробелы. В русском и во многих европейских языках общепринятым разделителем предложений является точка. В рассмотренном нами классном журнале в качестве разделителя можно использовать любой символ, который не встречается в самих данных, например символ «*». Тогда список выглядел бы так:

Аистов Александр Алексеевич * Бобров Борис Борисович * Воробьева Валентина Владиславовна * ... * Сорокин Сергей Семенович

В этом случае для розыска элемента с номером n надо просмотреть список начиная с самого начала и пересчитать встретившиеся разделители. Когда будет отсчитано $n-1$ разделителей, начнется нужный элемент. Он закончится, когда будет встречен следующий разделитель.

Еще проще можно действовать, если все элементы списка имеют равную длину. В этом случае разделители в списке вообще не нужны. Для розыска элемента

с номером n надо просмотреть список с самого начала и отсчитать $a(n-1)$ символ, где a — длина одного элемента. Со следующего символа начнется нужный элемент. Его длина тоже равна a , поэтому его конец определить нетрудно. Такие упрощенные списки, состоящие из элементов равной длины, называют *векторами данных*. Работать с ними особенно удобно.

Таким образом, *линейные структуры данных (списки)* — это упорядоченные структуры, в которых адрес элемента однозначно определяется его номером.

Табличные структуры (таблицы данных, матрицы данных)

С таблицами данных мы тоже хорошо знакомы, достаточно вспомнить всем известную таблицу умножения. Табличные структуры отличаются от списочных тем, что элементы данных определяются *адресом ячейки*, который состоит не из одного параметра, как в списках, а из нескольких. Для таблицы умножения, например, адрес ячейки определяется номерами строки и столбца. Нужная ячейка находится на их пересечении, а элемент выбирается из ячейки.

При хранении табличных данных количество разделителей должно быть больше, чем для данных, имеющих структуру списка. Например, когда таблицы печатают в книгах, строки и столбцы разделяют графическими элементами — линиями вертикальной и горизонтальной разметки (рис. 1.4).

Если нужно сохранить таблицу в виде длинной символьной строки, используют один символ-разделитель между элементами, принадлежащими одной строке, и другой разделитель для отделения строк, например так:

Меркурий*0,39*0,056*0#Венера*0,67*0,88*0#Земля*1,0*1,0*1#Марс*1,51*0,1*2#...

Для розыска элемента, имеющего адрес ячейки (m, n) , надо просмотреть набор данных с самого начала и пересчитать внешние разделители. Когда будет отсчитан $m-1$ разделитель, надо пересчитывать внутренние разделители. После того как будет найден $n-1$ разделитель, начнется нужный элемент. Он закончится, когда будет встречен любой очередной разделитель.

Планета	Расстояние от Солнца, а. е.	Относительная масса	Количество спутников
Меркурий	0,30	0,056	0
Венера	0,67	0,88	0
Земля	1,0	1,0	1
Марс	1,51	0,1	2
Юпитер	5,2	318	16

Рис. 1.4. В двумерных таблицах, которые печатают в книгах, применяется два типа разделителей — вертикальные и горизонтальные

Еще проще можно действовать, если все элементы таблицы имеют равную длину. Такие таблицы называют *матрицами*. В данном случае разделители не нужны, поскольку все элементы имеют равную длину и количество их известно. Для розыска элемента с адресом (m, n) в матрице, имеющей M строк и N столбцов,

надо просмотреть ее с самого начала и отсчитать $a[N(m-1) + (n-1)]$ символ, где a — длина одного элемента. Со следующего символа начнется нужный элемент. Его длина тоже равна a , поэтому его конец определить нетрудно.

Таким образом, табличные структуры данных (*матрицы*) — это упорядоченные структуры, в которых **адрес элемента определяется номером строки и номером столбца, на пересечении которых находится ячейка, содержащая искомый элемент.**

Многомерные таблицы. Выше мы рассмотрели пример таблицы, имеющей два измерения (строка и столбец), но в жизни нередко приходится иметь дело с таблицами, у которых количество измерений больше. Вот пример таблицы, с помощью которой может быть организован учет учащихся.

Номер факультета	3
Номер курса (на факультете)	2
Номер специальности (на курсе)	2
Номер группы в потоке одной специальности	1
Номер учащегося в группе	19

Размерность такой таблицы равна пяти, и для однозначного отыскания данных об учащемся в подобной структуре надо знать все пять параметров (координат).

Иерархические структуры данных

Нерегулярные данные, которые трудно представить в виде списка или таблицы, часто представляют в виде *иерархических структур*. С подобными структурами мы очень хорошо знакомы по обыденной жизни. Иерархическую структуру имеет система почтовых адресов. Подобные структуры также широко применяют в научных систематизациях и всевозможных классификациях (рис. 1.5).

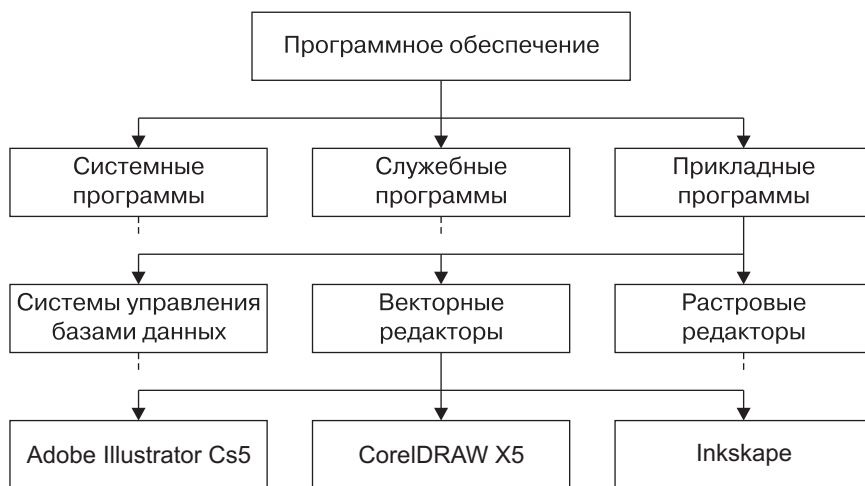


Рис. 1.5. Пример иерархической структуры данных

В иерархической структуре адрес каждого элемента определяется путем доступа (маршрутом), ведущим от вершины структуры к данному элементу. Вот, например, как выглядит путь доступа к команде, запускающей программу Калькулятор (стандартная программа компьютеров, работающих в операционной системе Windows 7):

Пуск ► Все программы ► Стандартные ► Калькулятор.

Дихотомия данных. Основным недостатком иерархических структур данных является увеличенный размер пути доступа. Очень часто бывает так, что длина маршрута оказывается больше, чем длина самих данных, к которым он ведет. Поэтому в информатике применяют методы для регуляризации иерархических структур с тем, чтобы сделать путь доступа компактным. Один из методов получил название *дихотомии*. Его суть понятна из примера, представленного на рис. 1.6.

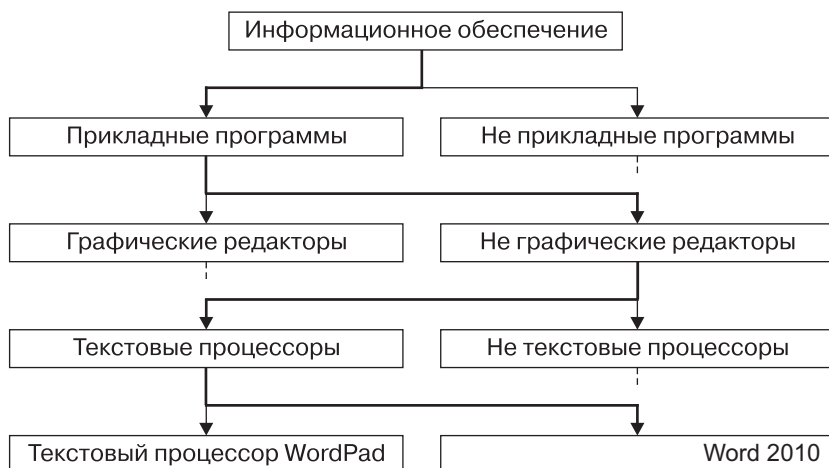


Рис. 1.6. Пример, поясняющий принцип действия метода дихотомии

В иерархической структуре, построенной методом дихотомии, путь доступа к любому элементу можно представить как путь через рациональный лабиринт с поворотами налево (0) или направо (1) и, таким образом, выразить путь доступа в виде компактной двоичной записи. В нашем примере путь доступа к текстовому процессору Microsoft Word 2010 выразится следующим двоичным числом: 1010.

Упорядочение структур данных

Списочные и табличные структуры являются простыми. Ими легко пользоваться, поскольку адрес каждого элемента задается числом (для списка), двумя числами (для двумерной таблицы) или несколькими числами для многомерной таблицы. Они также легко упорядочиваются. Основным методом упорядочения является *сортировка*. Данные можно сортировать по любому избранному критерию, например: по алфавиту, по возрастанию порядкового номера или по возрастанию какого-либо параметра.

Несмотря на многочисленные удобства, у простых структур данных есть и недостаток — их трудно обновлять. Если, например, перевести студента из одной группы в другую, изменения надо вносить сразу в два журнала посещаемости; при этом в обоих журналах будет нарушена списочная структура. Если переведенного студента вписать в конец списка группы, нарушится упорядочение по алфавиту, а если его вписать в соответствии с алфавитом, то изменятся порядковые номера всех студентов, которые следуют за ним.

Таким образом, **при добавлении произвольного элемента в упорядоченную структуру списка может происходить изменение адресных данных у других элементов.** В журналах успеваемости это пережить нетрудно, но в системах, выполняющих автоматическую обработку данных, нужны специальные методы для решения этой проблемы.

Иерархические структуры данных по форме сложнее, чем линейные и табличные, но они не создают проблем с обновлением данных. Их легко развивать путем создания новых уровней. Даже если в учебном заведении будет создан новый факультет, это никак не отразится на пути доступа к сведениям об учащихся прочих факультетов.

Недостатком иерархических структур является относительная трудоемкость записи адреса элемента данных и сложность упорядочения. Часто методы упорядочения в таких структурах основывают на предварительной *индексации*, которая заключается в том, что каждому элементу данных присваивается свой уникальный индекс, который можно использовать при поиске, сортировке и т. п. Ранее рассмотренный принцип дихотомии на самом деле является одним из методов индексации данных в иерархических структурах. После такой индексации данные легко разыскиваются по двоичному коду связанного с ними индекса.

Адресные данные. Если данные хранятся не как попало, а в организованной структуре (причем любой), то каждый элемент данных приобретает новое свойство (параметр), который можно назвать *адресом*. Конечно, работать с упорядоченными данными удобнее, но за это приходится платить их размножением, поскольку адреса элементов данных — это тоже данные и их тоже надо хранить и обрабатывать.

1.3. Файлы и файловая структура

Единицы представления данных

Существует множество систем представления данных. С одной из них, принятой в информатике и вычислительной технике, двоичным кодом, мы познакомились выше. Наименьшей единицей такого представления является бит (*двоичный разряд*).

Совокупность двоичных разрядов, выражающих числовые или иные данные, образует некий битовый рисунок. Практика показывает, что с битовым представлением удобнее работать, если этот рисунок имеет регулярную форму. В настоящее время в качестве таких форм используются группы из восьми битов, которые называются *байтами*.

Понятие о байте как группе взаимосвязанных битов появилось вместе с первыми образцами электронной вычислительной техники. Долгое время оно было

машиннозависимым, то есть для разных вычислительных машин длина байта была разной. Только в конце 60-х годов понятие байта стало универсальным и *машиннонезависимым*.

Десятичное число	Двоичное число	Байт
1	1	0000 0001
2	10	0000 0010
...		...
255	11111111	1111 1111

Выше мы видели, что во многих случаях целесообразно использовать не восьмиразрядное кодирование, а 16-разрядное, 24-разрядное, 32-разрядное и более. Группа из 16 взаимосвязанных бит (двух взаимосвязанных байтов) в информатике называется *словом*. Соответственно, группы из четырех взаимосвязанных байтов (32 разряда) называются *удвоенным словом*, группы из восьми байтов (64 разряда) — *четверным словом*, группы из шестнадцати байтов (128 разрядов) — *восьмерным словом*. Пока, на сегодняшний день, такой системы обозначения достаточно.

Единицы измерения данных

Существует много различных систем и единиц измерения данных. Каждая научная дисциплина и каждая область человеческой деятельности может использовать свои, наиболее удобные или традиционно устоявшиеся единицы. В информатике для измерения данных используют тот факт, что разные типы данных имеют универсальное двоичное представление и потому вводят свои единицы данных, основанные на нем.

Наименьшей единицей измерения является байт. Поскольку одним байтом, как правило, кодируется один символ текстовой информации, то для текстовых документов размер в байтах соответствует лексическому объему в символах (пока исключение представляет рассмотренная выше универсальная кодировка *UNICODE*).

Более крупная единица измерения — килобайт (Кбайт). Условно можно считать, что 1 Кбайт примерно равен 1000 байт. Условность связана с тем, что для вычислительной техники, работающей с двоичными числами, более удобно представление чисел в виде степени двойки и потому на самом деле 1 Кбайт равен 2^{10} байт (1024 байт). Однако всюду, где это не принципиально, с инженерной погрешностью (до 3 %) «забывают» о «лишних» байтах.

В килобайтах измеряют сравнительно небольшие объемы данных. Условно можно считать, что одна страница неформатированного машинописного текста составляет около двух килобайт.

Более крупные единицы измерения данных образуются добавлением префиксов *мега-*, *гига-*, *тера-*; в более крупных единицах пока нет практической надобности.

$$1 \text{ Мбайт} = 1024 \text{ Кбайт} = 10^{20} \text{ байт}$$

$$1 \text{ Гбайт} = 1024 \text{ Мбайт} = 10^{30} \text{ байт}$$

$$1 \text{ Тбайт} = 1024 \text{ Гбайт} = 10^{40} \text{ байт}$$

Особо обратим внимание на то, что при переходе к более крупным единицам «инженерная» погрешность, связанная с округлением, накапливается и становится недопустимой, поэтому на старших единицах измерения округление производится реже.

Единицы хранения данных

При хранении данных решаются две проблемы: как сохранить данные в наиболее компактном виде и как обеспечить к ним удобный и быстрый доступ (если доступ не обеспечен, то это не хранение). Для обеспечения доступа необходимо, чтобы данные имели упорядоченную структуру, а при этом, как мы уже знаем, образуется «паразитная нагрузка» в виде адресных данных. Без них нельзя получить доступ к нужным элементам данных, входящих в структуру.

Поскольку адресные данные тоже имеют размер и тоже подлежат хранению, хранить данные в виде мелких единиц, таких как байты, неудобно. Их неудобно хранить и в более крупных единицах (килобайтах, мегабайтах и т. п.), поскольку неполное заполнение одной единицы хранения приводит к неэффективности хранения.

В качестве единицы хранения данных принят объект переменной длины, называемый *файлом*. *Файл — это последовательность произвольного числа байтов, обладающая уникальным собственным именем*. Обычно в отдельном файле хранят данные, относящиеся к одному типу. В этом случае тип данных определяет *тип файла*.

Проще всего представить себе файл в виде безразмерного канцелярского досье, в которое можно по желанию добавлять содержимое или извлекать его оттуда. Поскольку в определении файла нет ограничений на размер, можно представить себе файл, имеющий 0 байт (*пустой файл*), и файл, имеющий любое число байтов.

В определении файла особое внимание уделяется имени. Оно фактически несет в себе адресные данные, без которых данные, хранящиеся в файле, не станут информацией из-за отсутствия метода доступа к ним. Кроме функций, связанных с адресацией, имя файла может хранить и сведения о типе данных, заключенных в нем. Для автоматических средств работы с данными это важно, поскольку по имени файла они могут автоматически определить адекватный метод извлечения информации из файла.

Понятие о файловой структуре

Требование уникальности имени файла очевидно — без этого невозможно гарантировать однозначность доступа к данным. В средствах вычислительной техники требование уникальности имени обеспечивается автоматически — создать файл с именем, тождественным уже имеющемуся, не может ни пользователь, ни автоматика.

Хранение файлов организуется в иерархической структуре, которая в данном случае называется *файловой структурой*. В качестве вершины структуры служит имя носителя, на котором сохраняются файлы. Далее файлы группируются в ка-

талог (*папки*), внутри которых могут быть созданы *вложенные каталоги* (*папки*). *Путь доступа к файлу* начинается с имени устройства и включает все имена каталогов (*папок*), через которые проходит. В качестве разделителя используется символ «\» (обратная косая черта).

Уникальность имени файла обеспечивается тем, что *полным именем файла считается собственное имя файла вместе с путем доступа к нему*. Понятно, что в этом случае на одном носителе не может быть двух файлов с тождественными полными именами.

Пример записи полного имени файла:

<имя носителя>\<имя каталога-1>\... \<имя каталога-N>\<собственное имя файла>

Вот пример записи двух файлов, имеющих одинаковое собственное имя и размещенных на одном носителе, но отличающихся путем доступа, то есть полным именем. Для наглядности имена каталогов (*папок*) напечатаны прописными буквами.

C:\АВТОМАТИЧЕСКИЕ АППАРАТЫ\ВЕНЕРА\АТМОСФЕРА\Результаты исследований

C:\РАДИОЛОКАЦИЯ\ВЕНЕРА\РЕЛЬЕФ\Результаты исследований

О том, как на практике реализуются файловые структуры, мы узнаем несколько позже, когда познакомимся со средствами вычислительной техники и с понятием *файловой системы*.

1.4. Информатика

Предмет и задачи информатики

Информатика — это техническая наука, систематизирующая приемы создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими.

Из этого определения видно, что информатика очень близка к технологии, поэтому ее предмет нередко называют *информационной технологией*.

Предмет информатики составляют следующие понятия:

- ☐ аппаратное обеспечение средств вычислительной техники;
- ☐ программное обеспечение средств вычислительной техники;
- ☐ средства взаимодействия аппаратного и программного обеспечения;
- ☐ средства взаимодействия человека с аппаратными и программными средствами.

Как видно из этого списка, в информатике особое внимание уделяется вопросам *взаимодействия*. Для этого даже есть специальное понятие — *интерфейс*. Методы и средства взаимодействия человека с аппаратными и программными средствами называют *пользовательским интерфейсом*. Соответственно, существуют *аппаратные интерфейсы*, *программные интерфейсы* и *аппаратно-программные интерфейсы*.

Основной задачей информатики является систематизация приемов и методов работы с аппаратными и программными средствами вычислительной техники. *Цель* систематизации состоит в выделении, внедрении и развитии передовых, наиболее

эффективных технологий, в автоматизации этапов работы с данными, а также в методическом обеспечении новых технологических исследований.

Информатика — практическая наука. Ее достижения должны проходить подтверждение практикой и приниматься в тех случаях, когда они соответствуют критерию повышения эффективности. В составе основной задачи информатики сегодня можно выделить следующие направления для практических приложений:

- ❑ архитектура вычислительных систем (приемы и методы построения систем, предназначенных для автоматической обработки данных);
- ❑ интерфейсы вычислительных систем (приемы и методы управления аппаратным и программным обеспечением);
- ❑ программирование (приемы, методы и средства разработки компьютерных программ);
- ❑ преобразование данных (приемы и методы преобразования структур данных);
- ❑ защита информации (обобщение приемов, разработка методов и средств защиты данных);
- ❑ автоматизация (функционирование программно-аппаратных средств без участия человека);
- ❑ стандартизация (обеспечение совместимости между аппаратными и программными средствами, а также между форматами представления данных, относящихся к различным типам вычислительных систем).

На всех этапах технического обеспечения информационных процессов для информатики ключевым понятием является *эффективность*. Для аппаратных средств под эффективностью понимают отношение производительности оборудования к его стоимости (с учетом стоимости эксплуатации и обслуживания). Для программного обеспечения под эффективностью понимают производительность лиц, работающих с ними (пользователей). В программировании под эффективностью понимают объем программного кода, создаваемого программистами в единицу времени.

В информатике все жестко ориентировано на эффективность. Вопрос, *как сделать ту или иную операцию*, для информатики является важным, но не основным. Основным же является вопрос, *как сделать данную операцию эффективно*.

Истоки и предпосылки информатики

Слово *информатика* происходит от французского слова *Informatique*, образованного в результате объединения терминов *Informacion* (*информация*) и *Automatique* (*автоматика*), что выражает ее суть как науки об автоматической обработке информации. Кроме Франции термин *информатика* используется в ряде стран Восточной Европы. В то же время в большинстве стран Западной Европы и США используется другой термин — *Computer Science* (*наука о средствах вычислительной техники*).

В качестве источников информатики обычно называют две науки — *документалистику* и *кибернетику*. Документалистика сформировалась в конце XIX века в связи с бурным развитием производственных отношений. Ее расцвет пришелся

на 20–30-е годы XX века, а основным предметом стало изучение рациональных средств и методов повышения эффективности документооборота.

Основы близкой к информатике технической науки *кибернетики* были заложены трудами по математической логике американского математика Норберта Винера, опубликованными в 1948 году, а само название происходит от греческого слова (*kyberneticos* — искусный в управлении).

Впервые термин *кибернетика* ввел французский физик Андре Мари Ампер в первой половине XIX века. Он занимался разработкой единой системы классификации всех наук и обозначил этим термином гипотетическую науку об управлении, которой в то время не существовало, но которая, по его мнению, должна была существовать.

Сегодня предметом кибернетики являются принципы построения и функционирования систем автоматического управления, а основными задачами — методы моделирования процесса принятия решений техническими средствами, связь между психологией человека и математической логикой, связь между информационным процессом отдельного индивидуума и информационными процессами в обществе, разработка принципов и методов искусственного интеллекта. На практике кибернетика во многих случаях опирается на те же программные и аппаратные средства вычислительной техники, что и информатика, а информатика, в свою очередь, заимствует у кибернетики математическую и логическую базу для развития этих средств.

Подведение итогов

Все процессы в природе сопровождаются *сигналами*. Зарегистрированные сигналы образуют *данные*. Данные преобразуются, транспортируются и потребляются с помощью *методов*. При взаимодействии данных и адекватных им методов образуется *информация*. Информация — это динамический объект, образующийся в ходе *информационного процесса*. Он отражает диалектическую связь между объективными данными и субъективными методами. Свойства информации зависят как от свойств данных, так и от свойств методов.

Данные различаются *типами*, что связано с различиями в физической природе сигналов, при регистрации которых образовались данные. В качестве средства хранения и транспортировки данных используются *носители данных*. Для удобства операций с данными их структурируют. Наиболее широко используются следующие структуры: *линейная, табличная и иерархическая* — они различаются методом адресации к данным. При сохранении данных образуются данные нового типа — *адресные данные*.

Вопросами систематизации приемов и методов создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники занимается техническая наука — *информатика*. С целью унификации приемов и методов работы с данными в вычислительной технике применяется универсальная система кодирования данных, называемая *двоичным кодом*. Элементарной единицей представления данных в двоичном коде является *двоичный разряд (бит)*. Другой, более крупной единицей представления данных является *байт*.

Основной единицей хранения данных является *файл*. Файл представляет собой последовательность байтов, имеющую собственное имя. Совокупность файлов образует файловую структуру, которая, как правило, относится к иерархическому типу. *Полный адрес* файла в файловой структуре является уникальным и включает в себя собственное имя файла и путь доступа к нему.

Вопросы для самоконтроля

1. Как вы можете объяснить бытовой термин «переизбыток информации»? Что имеется в виду: излишняя полнота данных; излишняя сложность методов; неадекватность поступающих данных и методов, имеющихся в наличии?
2. Как вы понимаете термин «средство массовой информации»? Что это? Средство массовой поставки данных? Средство, обеспечивающее массовое распространение методов? Средство, обеспечивающее процесс информирования путем поставки данных гражданам, обладающим адекватными методами их потребления?
3. Как вы полагаете, являются ли данные товаром? Могут ли методы быть товаром?
4. На примере коммерческих структур, обеспечивающих коммуникационные услуги, покажите, как взаимодействуют между собой маркетинг данных и маркетинг методов? Можете ли вы привести примеры лизинга данных и методов?
5. Как вы понимаете диалектическое единство данных и методов? Можете ли вы привести примеры аналогичного единства двух понятий из других научных дисциплин: естественных, социальных, технических?
6. Как вы понимаете динамический характер информации? Что происходит с ней по окончании информационного процесса?
7. Можем ли мы утверждать, что данные, полученные в результате информационного процесса, адекватны исходным? Почему? От каких свойств исходных данных и методов зависит адекватность результирующих данных?
8. Что такое *вектор данных*? Является ли список номеров телефонов населенного пункта вектором данных? Является ли вектором данных текстовый документ, закодированный двоичным кодом, если он не содержит элементов оформления?
9. Является ли цифровой код цветного фотоснимка вектором данных? Если нет, то чего ему не хватает?
10. Как вы понимаете следующие термины: *аппаратно-программный интерфейс*, *программный интерфейс*, *аппаратный интерфейс*? Как бы вы назвали специальность людей, разрабатывающих аппаратные интерфейсы? Как называется специальность людей, разрабатывающих программные интерфейсы?
11. На основе личных наблюдений сделайте вывод о том, какими средствами может пользоваться преподаватель для обеспечения интерфейса с аудиторией. Можете ли вы рассмотреть отдельно методические и технические средства, имеющиеся в его распоряжении? Может ли преподаватель рассматривать *вашу* тетрадь и авторучку как *свое* средство обеспечения интерфейса? Если да, то в какой мере?

Глава вторая

Вычислительная техника

2.1. История развития средств вычислительной техники

Вычислительная система, компьютер

Изыскание средств и методов механизации и автоматизации работ — одна из основных задач технических дисциплин. Автоматизация работ с данными имеет свои особенности и отличия от автоматизации других типов работ. Для этого класса задач используют особые виды устройств, большинство из которых являются электронными приборами. Совокупность устройств, предназначенных для автоматической или автоматизированной обработки данных, называют *вычислительной техникой*. Конкретный набор взаимодействующих между собой устройств и программ, предназначенный для обслуживания одного рабочего участка, называют *вычислительной системой*. Центральным устройством большинства вычислительных систем является *компьютер*.

Компьютер — это электронный прибор, предназначенный для автоматизации создания, хранения, обработки и транспортировки данных.

Принцип действия компьютера

В определении компьютера как прибора мы указали определяющий признак — *электронный*. Однако автоматические вычисления не всегда производились электронными устройствами. Известны и механические устройства, способные выполнять расчеты автоматически.

Анализируя раннюю историю вычислительной техники, некоторые зарубежные исследователи нередко в качестве древнего предшественника компьютера называют механическое счетное устройство *абак*. Подход «от абак» свидетельствует о глубоком методическом заблуждении, поскольку абак не обладает свойством автоматического выполнения вычислений, а для компьютера оно определяющее.

Абак — наиболее раннее счетное механическое устройство, первоначально представлявшее собой глиняную пластину с желобами, в которых раскладывались камни, представляющие числа. Появление абак относят к четвертому тысячелетию до н. э. Местом появления считается Азия. В средние века в Европе абак сменился разграфленными таблицами. Вычисления с их помощью называли *счетом на линиях*, а в России в XVI–XVII веках появилось намного более передовое изобретение, применяемое и поныне, — *русские счеты*.

В то же время нам хорошо знаком другой прибор, способный автоматически выполнять вычисления, — это часы. Независимо от принципа действия, все виды часов (песочные, водяные, механические, электрические, электронные и др.) обладают способностью генерировать через равные промежутки времени перемещения или сигналы и регистрировать возникающие при этом изменения, то есть выполнять автоматическое суммирование сигналов или перемещений. Этот

принцип прослеживается даже в солнечных часах, содержащих только устройство регистрации (роль генератора выполняет система Земля — Солнце).

Механические часы — прибор, состоящий из устройства, автоматически выполняющего перемещения через равные заданные интервалы времени и устройства регистрации этих перемещений. Место появления первых механических часов неизвестно. Наиболее ранние образцы относятся к XIV веку и принадлежат монастырям (*башенные часы*).

В основе любого современного компьютера, как и в электронных часах, лежит *тактовый генератор*, вырабатывающий через равные интервалы времени электрические сигналы, которые используются для приведения в действие всех устройств компьютерной системы. Управление компьютером фактически сводится к управлению распределением сигналов между устройствами. Такое управление может производиться автоматически (в этом случае говорят о *программном управлении*) или вручную с помощью внешних органов управления — кнопок, переключателей, перемычек и т. п. (в ранних моделях). В современных компьютерах внешнее управление в значительной степени автоматизировано с помощью специальных аппаратно-логических интерфейсов, к которым подключаются устройства управления и ввода данных (клавиатура, мышь, джойстик и др.). В отличие от программного управления такое управление называют *интерактивным*.

Механические первоисточники

Первое в мире автоматическое устройство для выполнения операции сложения было создано на базе механических часов. В 1623 году его разработал Вильгельм Шикард, профессор кафедры восточных языков в университете Тюбингена (Германия). В наши дни рабочая модель устройства была воспроизведена по чертежам и подтвердила свою работоспособность. Сам изобретатель в письмах называл машину «суммирующими часами».

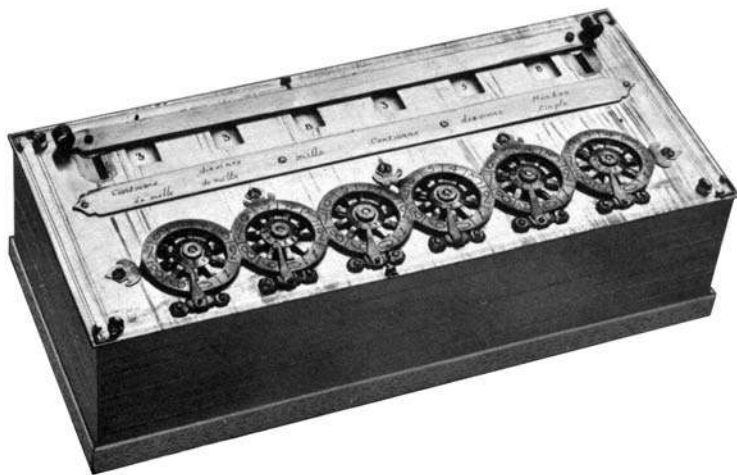


Рис. 2.1. Суммирующая машина Паскаля

В 1642 году французский механик Блез Паскаль (1623–1662) разработал более компактное суммирующее устройство (рис. 2.1), которое стало первым в мире механическим калькулятором, выпускавшимся серийно (главным образом для нужд парижских ростовщиков и менял). В 1673 году немецкий математик и философ Г. В. Лейбниц (1646–1717) создал механический калькулятор, который мог выполнять операции умножения и деления путем многократного повторения операций сложения и вычитания.

На протяжении XVIII века, известного как эпоха Просвещения, появились новые, более совершенные модели, но принцип механического управления вычислительными операциями оставался тем же. Идея программирования вычислительных операций пришла из той же часовой промышленности. Старинные монастырские башенные часы были настроены так, чтобы в заданное время включать механизм, связанный с системой колоколов. Такое программирование было *жестким* — одна и та же операция выполнялась в одно и то же время.

Идея гибкого программирования механических устройств с помощью перфорированной бумажной ленты впервые была реализована в 1804 году в ткацком станке Жаккарда, после чего оставался только один шаг до программного управления вычислительными операциями.

Этот шаг был сделан выдающимся английским математиком и изобретателем Чарльзом Бэббиджем (1792–1871) в его Аналитической машине, которая, к сожалению, так и не была до конца построена изобретателем при жизни, но была воспроизведена в наши дни по его чертежам, так что сегодня мы вправе говорить об Аналитической машине, как о реально существующем устройстве. Особенностью Аналитической машины стало то, что здесь впервые был реализован *принцип разделения информации на команды и данные*. Аналитическая машина содержала два крупных узла — «склад» и «мельницу». Данные вводились в механическую память «склада» путем установки блоков шестерен, а потом обрабатывались в «мельнице» с использованием команд, которые вводились с перфорированных карт (как в ткацком станке Жаккарда).



Рис. 2.2. Чарльз Бэббидж

Исследователи творчества Чарльза Бэббиджа непременно отмечают особую роль в разработке проекта Аналитической машины графини Огасты Ады Лавлейс (1815–1852), дочери известного поэта лорда Байрона. Именно ей принадлежала идея использования перфорированных карт для программирования вычислительных операций (1843). В частности, в одном из писем она писала: «Аналитическая машина точно так же плетет алгебраические узоры, как ткацкий станок воспроизводит цветы и листья». Леди Аду можно с полным основанием назвать самым первым в мире программистом. Сегодня ее именем назван один из известных языков программирования.

Идея Чарльза Бэббиджа о раздельном рассмотрении *команд* и *данных* оказалась необычайно плодотворной. В XX веке она была развита в принципах Джона фон Неймана (1941), и сегодня в вычислительной технике принцип раздельного рас-

смотрения *программ* и *данных* имеет очень важное значение. Он учитывается и при разработке архитектур современных компьютеров, и при разработке компьютерных программ.

Математические первоисточники

Если мы задумаемся над тем, с какими объектами работали первые механические предшественники современного электронного компьютера, то должны признать, что числа представлялись либо в виде линейных перемещений цепных и реечных механизмов, либо в виде угловых перемещений зубчатых и рычажных механизмов. И в том и в другом случае это были перемещения, что не могло не сказываться на габаритах устройств и на скорости их работы. Только переход от регистрации перемещений к регистрации сигналов позволил значительно снизить габариты и повысить быстроту действия. Однако на пути к этому достижению потребовалось ввести еще несколько важных принципов и понятий.

Двоичная система Лейбница. В механических устройствах зубчатые колеса могут иметь достаточно много фиксированных и, главное, *различимых* между собой положений. Количество таких положений, по крайней мере, равно числу зубьев шестерни. В электрических и электронных устройствах речь идет не о регистрации *положений* элементов конструкции, а о регистрации *состояний* элементов устройства. Таких устойчивых и *различимых* состояний всего два: включен — выключен; открыт — закрыт; заряжен — разряжен и т. п. Поэтому традиционная десятичная система, использованная в механических калькуляторах, неудобна для электронных вычислительных устройств.



Рис. 2.3. Готфрид Вильгельм Лейбниц

Возможность представления любых чисел (да и не только чисел) двоичными цифрами впервые была предложена Готфридом Вильгельмом Лейбницем в 1666 году. Он пришел к двоичной системе счисления, занимаясь исследованиями философской концепции единства и борьбы противоположностей. Попытка представить мироздание в виде непрерывного взаимодействия двух начал («черного» и «белого», мужского и женского, добра и зла) и применить к его изучению методы «чистой» математики подтолкнули Лейбница к изучению свойств двоичного представления данных с помощью нулей и единиц. Надо сказать, что Лейбницу уже тогда приходила в голову мысль о возможности использования двоичной системы в вычислительном устройстве, но, поскольку для механических устройств в этом не было никакой необходимости, он не стал использовать в своем калькуляторе (1673) принципы двоичной системы.

Математическая логика Джорджа Буля. Говоря о творчестве Джорджа Буля, исследователи истории вычислительной техники непременно подчеркивают, что этот выдающийся английский ученый первой половины XIX века был самоучкой. Возможно, именно благодаря отсутствию «классического» (в понимании того времени) образования Джордж Буль внес в логику как в науку революционные изменения.

Занимаясь исследованием законов мышления, он применил в логике систему формальных обозначений и правил, близкую к математической. Впоследствии эту систему назвали *логической алгеброй* или *булевой алгеброй*. Правила этой системы применимы к самым разнообразным объектам и их группам (*множествам*, по терминологии автора). Основное назначение системы, по замыслу Дж. Буля, состояло в том, чтобы кодировать логические высказывания и сводить структуры логических умозаключений к простым выражениям, близким по форме к математическим формулам. Результатом формального расчета логического выражения является одно из двух логических значений: *истина* или *ложь*.



Рис. 2.4. Джордж Буль

Значение логической алгебры долгое время игнорировалось, поскольку ее приемы и методы не содержали практической пользы для науки и техники того времени. Однако, когда появилась принципиальная возможность создания средств вычислительной техники на электронной базе, операции, введенные Булем, оказались весьма полезны. Они изначально ориентированы на работу только с двумя сущностями: *истина* и *ложь*. Нетрудно понять, как они пригодились для работы с двоичным кодом, который в современных компьютерах тоже представляется всего двумя сигналами: *ноль* и *единица*.

Не вся система Джорджа Буля (как и не все предложенные им логические операции) была использована при создании электронных вычислительных машин, но четыре основные операции (рис. 2.5): И (пересечение), ИЛИ (объединение), НЕ (обращение) и ИСКЛЮЧАЮЩЕЕ ИЛИ — лежат в основе работы всех видов процессоров современных компьютеров.

Операнды	И	или	НЕ (один операнд)	Исключающее или

Рис. 2.5. Основные операции логической алгебры

2.2. Методы классификации компьютеров

Существует достаточно много систем классификации компьютеров. Мы рассмотрим лишь некоторые из них, сосредоточившись на тех, о которых наиболее часто упоминают в доступной технической литературе и средствах массовой информации.

Классификация по назначению

Классификация по назначению — один из наиболее ранних методов классификации. Он связан с тем, как компьютер применяется. По этому принципу различают *большие ЭВМ* (*электронно-вычислительные машины*), *мини-ЭВМ*, *микро-ЭВМ* и *персональные компьютеры*, которые, в свою очередь, подразделяют на *массовые*, *деловые*, *портативные*, *развлекательные* и *рабочие станции*.

К 2010 году класс больших ЭВМ представлен в основном суперкомпьютерами, мини-ЭВМ и микро-ЭВМ включают в себя универсальные серверы различной мощности.

Большие ЭВМ. Это самые мощные компьютеры. Их применяют для обслуживания очень крупных организаций и даже целых отраслей народного хозяйства. За рубежом компьютеры этого класса называют *мэйнфреймами (mainframe)*. В России за ними закрепился термин *большие ЭВМ*. Штат обслуживания большой ЭВМ достигает многих десятков человек. На базе таких суперкомпьютеров создают *вычислительные центры* (рис. 2.6), включающие в себя несколько отделов или групп.

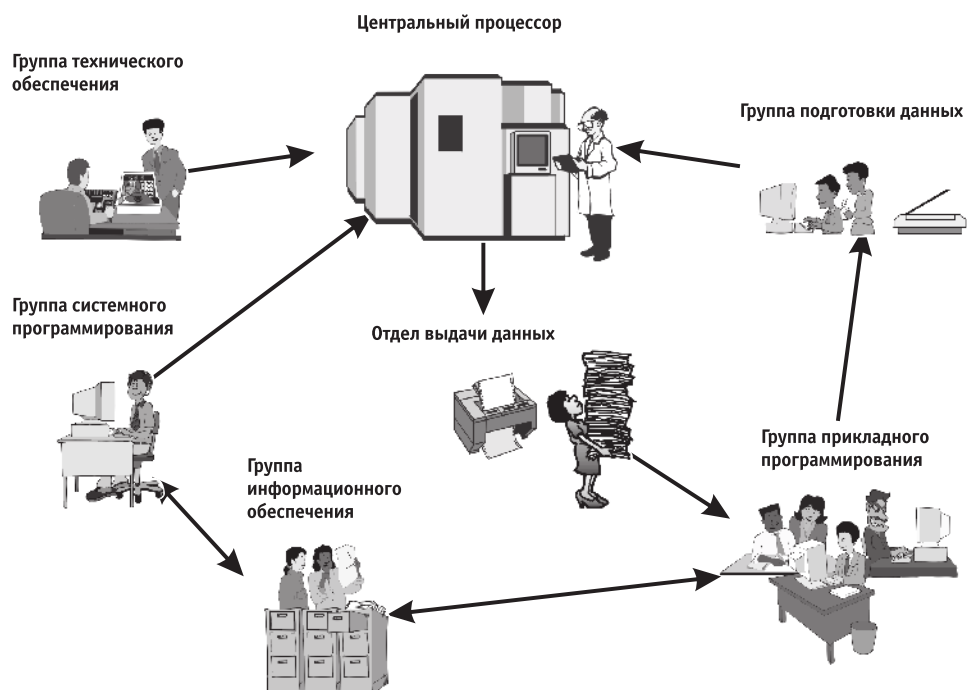


Рис. 2.6. Структура современного вычислительного центра на базе большой ЭВМ

Центральный процессор — основной блок ЭВМ, в котором непосредственно и происходит обработка данных и вычисление результатов. Обычно центральный процессор представляет собой несколько стоек аппаратуры и размещается в отдельном помещении, в котором соблюдаются повышенные требования по температуре, влажности, защищенности от электромагнитных помех, пыли и дыма.

Группа системного программирования занимается разработкой, отладкой и внедрением программного обеспечения, необходимого для функционирования самой вычислительной системы. Работников этой группы называют *системными программистами*. Они должны хорошо знать техническое устройство всех компонентов ЭВМ, поскольку их программы предназначены в первую очередь для управления физическими устройствами. Системные программы обеспечивают взаимодействие

программ более высокого уровня с оборудованием, то есть группа системного программирования обеспечивает *программно-аппаратный интерфейс* вычислительной системы.

Группа прикладного программирования занимается созданием программ для выполнения конкретных операций с данными. Работников этой группы называют *прикладными программистами*. В отличие от системных программистов им не надо знать техническое устройство компонентов ЭВМ, поскольку их программы работают не с устройствами, а с программами, подготовленными системными программистами. С другой стороны, с их программами работают пользователи, то есть конкретные исполнители работ. Поэтому можно говорить о том, что группа прикладного программирования обеспечивает *пользовательский интерфейс* вычислительной системы.

Группа подготовки данных занимается подготовкой данных, с которыми будут работать программы, созданные прикладными программистами. Во многих случаях сотрудники этой группы сами вводят данные с помощью клавиатуры, но они могут выполнять и преобразование готовых данных из одного вида в другой. Например, они могут получать иллюстрации, нарисованные художниками на бумаге, и преобразовывать их в электронный вид с помощью специальных устройств, называемых *сканерами*.

Группа технического обеспечения занимается техническим обслуживанием всей вычислительной системы, ремонтом и наладкой устройств, а также подключением новых устройств, необходимых для работы прочих подразделений.

Группа информационного обеспечения обеспечивает технической информацией все прочие подразделения вычислительного центра по их заказу. Эта же группа создает и хранит архивы ранее разработанных программ и накопленных данных. Такие архивы называют *библиотеками программ* или *банками данных*.

Отдел выдачи данных получает данные от центрального процессора и преобразует их в форму, удобную для заказчика. Здесь информация распечатывается на печатающих устройствах (*принтерах*) или отображается на экранах дисплеев.

Большие ЭВМ отличаются высокой стоимостью оборудования и обслуживания, поэтому работа таких суперкомпьютеров организована по непрерывному циклу. Наиболее трудоемкие и продолжительные вычисления планируют на ночные часы, когда количество обслуживающего персонала минимально. В дневное время ЭВМ исполняет менее трудоемкие, но более многочисленные задачи. При этом для повышения эффективности компьютер работает одновременно с несколькими задачами и, соответственно, с несколькими пользователями. Он поочередно переключается с одной задачи на другую и делает это настолько быстро и часто, что у каждого пользователя создается впечатление, будто компьютер работает только с ним. Такое распределение ресурсов вычислительной системы носит название *принципа разделения времени*.

Мини-ЭВМ

От больших ЭВМ компьютеры этой группы отличаются уменьшенными размерами и, соответственно, меньшей производительностью и стоимостью. Такие

компьютеры используются крупными предприятиями, научными учреждениями и некоторыми высшими учебными заведениями, сочетающими учебную деятельность с научной.

Мини-ЭВМ часто применяют для управления производственными процессами. Например, в механическом цехе компьютер может поддерживать ритмичность подачи заготовок, узлов и комплектующих на рабочие места; управлять гибкими автоматизированными линиями и промышленными роботами; собирать информацию с инструментальных постов технического контроля и сигнализировать о необходимости замены изношенных инструментов и приспособлений; готовить данные для станков с числовым программным управлением; а также своевременно информировать цеховые и заводские службы о необходимости выполнения мероприятий по переналадке оборудования.

Тот же компьютер может сочетать управление производством с другими задачами. Например, он может помогать экономистам в осуществлении контроля над себестоимостью продукции, нормировщикам в оптимизации времени технологических операций, конструкторам в автоматизации проектирования станочных приспособлений, бухгалтерии в осуществлении учета первичных документов и подготовки регулярных отчетов для налоговых органов. Для организации работы с мини-ЭВМ тоже требуется специальный вычислительный центр, хотя и не такой многочисленный, как для больших ЭВМ.

Микро-ЭВМ

Компьютеры данного класса доступны многим предприятиям. Организации, использующие микро-ЭВМ, обычно не создают вычислительные центры. Для обслуживания такого компьютера им достаточно небольшой вычислительной лаборатории в составе нескольких человек. В число сотрудников вычислительной лаборатории обязательно входят программисты, хотя напрямую разработкой программ они не занимаются. Необходимые системные программы обычно покупают вместе с микро-ЭВМ, а разработку нужных прикладных программ заказывают более крупным вычислительным центрам или специализированным организациям.

Программисты вычислительной лаборатории занимаются внедрением приобретенного или заказанного программного обеспечения, выполняют его доводку и настройку, согласовывают его работу с другими программами и устройствами компьютера. Хотя программисты этой категории и не разрабатывают системные и прикладные программы, они могут вносить в них изменения, создавать или изменять отдельные фрагменты. Это требует высокой квалификации и универсальных знаний. Программисты, обслуживающие микро-ЭВМ, часто сочетают в себе качества системных и прикладных программистов одновременно.

Несмотря на относительно невысокую производительность по сравнению с большими ЭВМ, микро-ЭВМ находят применение и в крупных вычислительных центрах. Там им поручают вспомогательные операции, для которых нет смысла использовать дорогие суперкомпьютеры. К таким задачам, например, относится предварительная подготовка данных.

Персональные компьютеры

Эта категория компьютеров получила особо бурное развитие в течение последних двадцати лет. Из названия видно, что такой компьютер предназначен для обслуживания одного рабочего места. Как правило, с персональным компьютером (ПК) работает один человек. Несмотря на свои небольшие размеры и относительно невысокую стоимость, современные ПК обладают немалой производительностью. Многие современные персональные модели превосходят большие ЭВМ 70-х годов, мини-ЭВМ 80-х годов и микро-ЭВМ первой половины 90-х годов. Персональный компьютер (*Personal Computer, PC*) вполне способен удовлетворить большинство потребностей малых предприятий и отдельных лиц.

Особенно широкую популярность ПК получили после 1995 года в связи с бурным развитием Интернета. Персонального компьютера вполне достаточно для использования всемирной сети в качестве источника научной, справочной, учебной, культурной и развлекательной информации. Персональные компьютеры являются также удобным средством автоматизации учебного процесса по любым дисциплинам, средством организации дистанционного (заочного) обучения и средством организации досуга. Они вносят большой вклад не только в производственные, но и в социальные отношения. Их нередко используют для организации домашней трудовой деятельности, что особенно важно в условиях ограниченной трудозанятости.

До последнего времени модели ПК условно рассматривали в двух категориях: *бытовые ПК* и *профессиональные ПК*. Бытовые модели, как правило, имели меньшую производительность, но в них были приняты особые меры для работы с цветной графикой и звуком, чего не требовалось для профессиональных моделей. В связи с достигнутым в последние годы резким удешевлением средств вычислительной техники границы между профессиональными и бытовыми моделями в значительной степени стерлись, и сегодня в качестве бытовых нередко используют высокопроизводительные профессиональные модели, а профессиональные модели, в свою очередь, комплектуют устройствами для воспроизведения мультимедийной информации, что ранее было характерно для бытовых устройств.

Под термином *мультимедиа* подразумевается сочетание нескольких видов данных в одном документе (текстовые, графические, музыкальные и видеоданные) или совокупность устройств для воспроизведения этого комплекса данных.

С 1999 по 2001 год в области ПК действовали международные сертификационные стандарты — *спецификации PC99–PC2001*. Они регламентировали принципы классификации ПК и оговаривали минимальные и рекомендуемые требования к каждой из категорий. Стандарты устанавливали следующие категории ПК:

- *Consumer PC* (массовый ПК);
- *Office PC* (деловой ПК);
- *Mobile PC* (портативный ПК);

- ❑ *Workstation PC* (рабочая станция);
- ❑ *Entertainment PC* (развлекательный ПК).

Каждая категория имела свои особенности: для *портативных ПК* обязательным было наличие средств компьютерной связи, в категории *рабочих станций* предъявлялись повышенные требования к устройствам хранения данных, а в категории *развлекательных ПК* — к средствам воспроизведения графики и звука.

Одна из целей такой стандартизации состояла и в том, чтобы наметить пути дальнейшего развития и совершенствования ПК. Однако развитие аппаратных средств ПК привело к постепенному размыванию границ между разными категориями, а планы развития часто не оправдывались. Поэтому обновление этих стандартов было прекращено, хотя при приобретении компьютера для конкретных задач классификацию, введенную этими стандартами, все еще полезно держать в голове.

Другие виды классификации компьютеров

Классификация по уровню специализации. По уровню специализации компьютеры делят на *универсальные* и *специализированные*. На базе универсальных компьютеров можно собирать вычислительные системы произвольного состава (состав компьютерной системы называется *конфигурацией*). Так, например, один и тот же ПК можно использовать для работы с текстами, музыкой, графикой, фото- и видеоматериалами.

Специализированные компьютеры предназначены для решения конкретного круга задач. К таким компьютерам относятся, например, бортовые компьютеры автомобилей, судов, самолетов, космических аппаратов. Бортовые компьютеры управляют средствами ориентации и навигации, осуществляют контроль состояния бортовых систем, выполняют некоторые функции автоматического управления и связи, а также большинство функций по оптимизации параметров работы систем объекта (например, оптимизацию расхода топлива в зависимости от конкретных условий движения объекта). Специализированные мини-ЭВМ, ориентированные на работу с графикой, называют *графическими станциями*. Их используют при подготовке кино- и видеофильмов, а также рекламной продукции. Специализированные компьютеры, объединяющие компьютеры предприятия в одну сеть, называют *файловыми серверами*. Компьютеры, обеспечивающие передачу информации между различными участниками всемирной компьютерной сети, называют *сетевыми серверами*.

Во многих случаях с задачами специализированных компьютерных систем могут справляться и обычные универсальные компьютеры, но считается, что использование специализированных систем все-таки эффективнее. Критерием оценки эффективности выступает отношение производительности оборудования к величине его стоимости.

Классификация по типоразмерам. Персональные компьютеры можно классифицировать по типоразмерам. Так, различают *настольные (desktop)*, *портативные (notebook)* и *карманные (palmtop)* модели.

Настольные модели распространены наиболее широко. Они являются принадлежностью рабочего места. Эти модели отличаются простотой изменения конфигурации за счет несложного подключения дополнительных внешних приборов или установки дополнительных внутренних компонентов. Достаточные размеры корпуса в настольном исполнении позволяют выполнять большинство подобных работ без привлечения специалистов, а это позволяет настраивать компьютерную систему оптимально для решения именно тех задач, для которых она была приобретена.

Портативные модели удобны для транспортировки. Благодаря сближению стоимости портативных и настольных моделей сегодня их применяют буквально все категории пользователей — от руководителей предприятий и организаций до студентов и школьников. С портативным компьютером можно работать при отсутствии рабочего места. Особая привлекательность портативных компьютеров связана с тем, что их можно использовать в качестве средства связи. Портативный компьютер можно подключить к мобильному телефону, к обычной телефонной сети или к беспроводной компьютерной сети. Это дает возможность выхода в Интернет практически из любой географической точки.

Для эксплуатации на рабочем месте портативные компьютеры не очень удобны, но их можно подключать к настольным компьютерам, используемым стационарно.

Карманные модели выполняют функции «интеллектуальных записных книжек». Они позволяют хранить оперативные данные и получать к ним быстрый доступ. Некоторые карманные модели имеют жестко встроенное программное обеспечение, что облегчает непосредственную работу, но снижает гибкость в выборе прикладных программ.

Мобильные вычислительные устройства сочетают в себе функции карманных моделей компьютеров и средств мобильной связи (сотовых радиотелефонов). Их отличительная особенность — возможность мобильной работы с Интернетом. Некоторые модели поддерживают возможность приема телевизионных передач. Дополнительно МВУ комплектуют средствами связи по инфракрасному лучу, по каналам радиосвязи, благодаря которым эти карманные устройства могут обмениваться данными с настольными ПК и друг с другом.

Классификация по совместимости. В мире существует множество различных видов и типов компьютеров. Они выпускаются разными производителями, собираются из разных деталей, работают с разными программами. При этом очень важным вопросом становится *совместимость* различных компьютеров между собой. От совместимости зависит взаимозаменяемость узлов и приборов, предназначенных для разных компьютеров, возможность переноса программ с одного компьютера на другой и возможность совместной работы разных типов компьютеров с одними и теми же данными.

Аппаратная совместимость. По аппаратной совместимости различают так называемые *аппаратные платформы*. В области персональных компьютеров сегодня наиболее широко распространены две аппаратные платформы — *IBM PC* и *Apple Macintosh*. Кроме них существуют и другие платформы, распространенность которых ограничивается отдельными регионами или отдельными отраслями. Принад-

лежность компьютеров к одной аппаратной платформе повышает совместимость между ними, а принадлежность к разным платформам — понижает.

Кроме аппаратной совместимости существуют и другие виды совместимости: *совместимость на уровне операционной системы, программная совместимость, совместимость на уровне данных*.

Классификация по типу используемого процессора. *Процессор* — основной компонент любого компьютера. В электронно-вычислительных машинах это специальный блок, а в персональных компьютерах — специальная микросхема, которая выполняет все вычисления в компьютере. Даже если компьютеры принадлежат одной аппаратной платформе, они могут различаться по типу используемого процессора. Основные типы процессоров для платформы *IBM PC* мы рассмотрим в соответствующем разделе, а здесь укажем на то, что тип используемого процессора в значительной (хотя и не в полной) мере характеризует технические свойства компьютера.

2.3. Состав вычислительной системы

Состав вычислительной системы называется *конфигурацией*. Аппаратные и программные средства вычислительной техники принято рассматривать отдельно. Соответственно, отдельно рассматривают *аппаратную конфигурацию* вычислительных систем и их *программную конфигурацию*. Такой принцип разделения имеет для информатики особое значение, поскольку очень часто решение одних и тех же задач может обеспечиваться как аппаратными, так и программными средствами. Критериями выбора аппаратного или программного решения являются производительность и эффективность.

Принято считать, что аппаратные решения в среднем оказываются дороже, зато реализация программных решений требует более высокой квалификации персонала.

Аппаратное обеспечение

К *аппаратному обеспечению* вычислительных систем относятся устройства и приборы, образующие аппаратную конфигурацию. Современные компьютеры и вычислительные комплексы имеют блочно-модульную конструкцию — аппаратную конфигурацию, необходимую для исполнения конкретных видов работ, можно собирать из готовых узлов и блоков.

По способу расположения устройств относительно *центрального процессорного устройства* (ЦПУ — *Central Processing Unit, CPU*) различают *внутренние* и *внешние* устройства. Внешними, как правило, являются большинство устройств ввода-вывода данных (их также называют *периферийными* устройствами) и некоторые устройства, предназначенные для длительного хранения данных.

Согласование между отдельными узлами и блоками выполняют с помощью переходных аппаратно-логических устройств, называемых *аппаратными интерфейсами*. Стандарты на аппаратные интерфейсы в вычислительной технике называют

протоколами. Таким образом, *протокол — это совокупность технических условий, которые должны быть обеспечены разработчиками устройств для успешного согласования их работы с другими устройствами.*

Многочисленные интерфейсы, присутствующие в архитектуре любой вычислительной системы, можно условно разделить на две большие группы: *последовательные и параллельные.* Через последовательный интерфейс данные передаются последовательно, бит за битом, а через параллельный — одновременно группами битов. Количество битов, участвующих в одной посылке, определяется разрядностью интерфейса, например восьмиразрядные параллельные интерфейсы передают один байт (8 бит) за один цикл.

Параллельные интерфейсы обычно имеют более сложное устройство, чем последовательные, но обеспечивают более высокую производительность. Их применяют там, где важна скорость передачи данных: для подключения печатающих устройств, устройств ввода графической информации, устройств записи данных на внешний носитель и т. п. Производительность параллельных интерфейсов измеряют *байтами в секунду* (байт/с; Кбайт/с; Мбайт/с).

Устройство последовательных интерфейсов проще; как правило, для них не надо синхронизировать работу передающего и принимающего устройства (поэтому их часто называют *асинхронными интерфейсами*). Первоначально пропускная способность последовательных интерфейсов была меньше, а коэффициент полезного действия — ниже. Из-за отсутствия синхронизации посылок полезные данные предваряют и завершают посылками служебных данных, то есть на один байт полезных данных могут приходиться 1–3 служебных бита (состав и структуру посылки определяет конкретный протокол).

Поскольку обмен данными через последовательные устройства производится не байтами, а битами, их производительность измеряют битами в секунду (бит/с, Кбит/с, Мбит/с). Несмотря на кажущуюся простоту перевода единиц измерения скорости последовательной передачи в единицы измерения скорости параллельной передачи данных путем механического деления на 8, такой пересчет не выполняют, поскольку он не корректен из-за наличия служебных данных. В крайнем случае, с поправкой на служебные данные, иногда скорость последовательных устройств выражают в *знаках в секунду* или, что то же самое, в *символах в секунду* (с/с), но эта величина имеет не технический, а справочный, потребительский характер.

Первоначально последовательные интерфейсы применяли для подключения «медленных» устройств (простейших устройств печати низкого качества, устройств ввода и вывода знаковой и сигнальной информации, контрольных датчиков, малопроизводительных устройств связи и т. п.), а также в тех случаях, когда отсутствуют существенные ограничения по продолжительности обмена данными.

Однако с развитием техники появились новые, высокоскоростные последовательные интерфейсы, не уступающие параллельным, а нередко и превосходящие их по пропускной способности. Сегодня последовательные интерфейсы применяют для подключения к компьютеру любых типов устройств.

Эволюция аппаратных интерфейсов приводит к постепенному вытеснению параллельных интерфейсов последовательными.

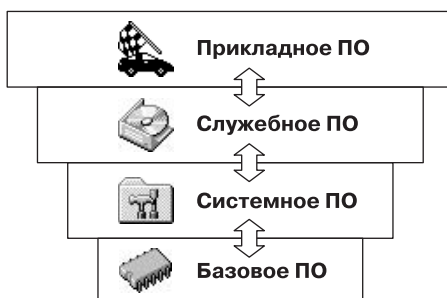
Программное обеспечение

Программы — это упорядоченные последовательности команд. Конечная цель любой компьютерной программы — управление аппаратными средствами. Даже если на первый взгляд программа никак не взаимодействует с оборудованием, не требует никакого ввода данных с устройств ввода и не осуществляет вывод данных на устройства вывода, все равно ее работа основана на управлении аппаратными устройствами компьютера.

Программное и аппаратное обеспечение в компьютере работают в неразрывной связи и в непрерывном взаимодействии. Несмотря на то что мы рассматриваем эти две категории отдельно, нельзя забывать, что между ними существует диалектическая связь и раздельное их рассмотрение является по меньшей мере условным.

Состав программного обеспечения вычислительной системы называют *программной конфигурацией*. Между программами, как и между физическими узлами и блоками, существует взаимосвязь — многие программы работают, опираясь на другие программы более низкого уровня, то есть мы можем говорить о межпрограммном *интерфейсе*. Возможность существования такого интерфейса тоже основана на существовании технических условий и протоколов взаимодействия, а на практике он обеспечивается распределением программного обеспечения на несколько взаимодействующих между собой уровней.

Уровни программного обеспечения представляют собой пирамидальную конструкцию. Каждый следующий уровень опирается на программное обеспечение предшествующих уровней. Такое членение удобно для всех этапов работы с вычислительной системой, начиная с установки программ до практической эксплуатации и технического обслуживания. Обратите внимание на то, что каждый вышележащий уровень повышает функциональность всей системы. Так, например, вычислительная система с программным обеспечением базового уровня не способна выполнять большинство функций, но позволяет установить системное программное обеспечение.



Базовый уровень. Самый низкий уровень программного обеспечения представляет *базовое программное обеспечение*. Оно отвечает за взаимодействие с *базовыми аппаратными средствами*. Как правило, базовые программные средства непосредственно входят в состав базового оборудования и хранятся в специальных

микросхемах, называемых *постоянными запоминающими устройствами* (ПЗУ — *Read Only Memory, ROM*). Программы и данные записываются («прошиваются») в микросхемы ПЗУ на этапе производства и не могут быть изменены в процессе эксплуатации.

В тех случаях, когда изменение базовых программных средств во время эксплуатации является технически целесообразным, вместо микросхем ПЗУ применяют *перепрограммируемые постоянные запоминающие устройства* (ППЗУ — *Erasable and Programmable Read Only Memory, EPROM*). В этом случае изменение содержания ПЗУ можно выполнять как непосредственно в составе вычислительной системы (такая технология называется *флэш-технологией*), так и вне нее, на специальных устройствах, называемых *программаторами*.

Системный уровень. Системный уровень — переходный. Программы, работающие на этом уровне, обеспечивают взаимодействие прочих программ компьютерной системы с программами базового уровня и непосредственно с аппаратным обеспечением, то есть выполняют «посреднические» функции.

От программного обеспечения этого уровня во многом зависят эксплуатационные показатели всей вычислительной системы в целом. Так, например, при подключении к вычислительной системе нового оборудования на системном уровне должна быть установлена программа, обеспечивающая для других программ взаимосвязь с этим оборудованием. Конкретные программы, отвечающие за взаимодействие с конкретными устройствами, называются *драйверами устройств* — они входят в состав программного обеспечения системного уровня.

Другой класс программ системного уровня отвечает за взаимодействие с пользователем. Именно благодаря им он получает возможность вводить данные в вычислительную систему, управлять ее работой и получать результат в удобной для себя форме. Эти программные средства называют *средствами обеспечения пользовательского интерфейса*. От них напрямую зависит удобство работы с компьютером и производительность труда на рабочем месте.

Совокупность программного обеспечения системного уровня образует *ядро операционной системы компьютера*. Полное понятие операционной системы мы рассмотрим несколько позже, а здесь только отметим, что если компьютер оснащен программным обеспечением системного уровня, то он уже подготовлен к установке программ более высоких уровней, к взаимодействию программных средств с оборудованием и, самое главное, к взаимодействию с пользователем. То есть *наличие ядра операционной системы — неперенное условие для возможности практической работы человека с вычислительной системой*.

Служебный уровень. Программное обеспечение этого уровня взаимодействует как с программами базового уровня, так и с программами системного уровня. Основное назначение служебных программ (их также называют *утилитами*) состоит в автоматизации работ по проверке, наладке и настройке компьютерной системы. Во многих случаях они используются для расширения или улучшения функций системных программ. Некоторые служебные программы (как правило, это программы обслуживания) изначально включают в состав операционной системы, но большинство служебных программ являются для операционной системы внешними и служат для расширения ее функций.

В разработке и эксплуатации служебных программ существует два альтернативных направления: *интеграция с операционной системой* и *автономное функционирование*. В первом случае служебные программы могут изменять потребительские свойства системных программ, делая их более удобными для практической работы. Во втором случае они слабо связаны с системным программным обеспечением, но предоставляют пользователю больше возможностей для персональной настройки их взаимодействия с аппаратным и программным обеспечением.

Прикладной уровень. Программное обеспечение прикладного уровня представляет собой комплекс прикладных программ, с помощью которых на данном рабочем месте выполняются конкретные задания. Спектр этих заданий необычайно широк: от производственных до творческих и развлекательно-обучающих. Огромный функциональный диапазон возможных приложений средств вычислительной техники обусловлен наличием прикладных программ для разных видов деятельности.

Поскольку между прикладным программным обеспечением и системным существует непосредственная взаимосвязь (первое опирается на второе), то можно утверждать, что универсальность вычислительной системы, доступность прикладного программного обеспечения и широта функциональных возможностей компьютера напрямую зависят от типа используемой операционной системы, от того, какие системные средства содержит ее ядро, как она обеспечивает взаимодействие триединого комплекса *человек — программы — оборудование*.

Классификация прикладных программных средств

Текстовые редакторы. Основные функции этого класса прикладных программ заключаются во вводе и редактировании текстовых данных. Дополнительные функции состоят в автоматизации процессов ввода и редактирования. Для операций ввода, вывода и сохранения данных текстовые редакторы вызывают и используют системное программное обеспечение. Впрочем, это характерно и для всех прочих видов прикладных программ, и в дальнейшем мы не будем специально указывать на этот факт. С этого класса прикладных программ обычно начинают знакомство с программным обеспечением и на нем отрабатывают первичные навыки взаимодействия с компьютерной системой.

Текстовые процессоры. Основное отличие текстовых процессоров от текстовых редакторов в том, что они позволяют не только вводить и редактировать текст, но и *форматировать* его, то есть оформлять. Соответственно, к основным средствам текстовых процессоров относятся средства обеспечения взаимодействия текста, графики, таблиц и других объектов, составляющих итоговый документ, а к дополнительным — средства автоматизации процесса форматирования.

Современный стиль работы с документами подразумевает два альтернативных подхода — работу с бумажными документами и работу с электронными документами (по безбумажной технологии). Поэтому, говоря о форматировании документов средствами текстовых процессоров, надо иметь в виду два принципиально разных направления — форматирование документов, предназначенных для печати, и форматирование электронных документов, предназначенных для отображения на экране. Приемы и методы в этих случаях существенно различаются. Соответственно,

различаются и текстовые процессоры, хотя многие из них успешно сочетают оба подхода.

Графические редакторы. Это обширный класс программ, предназначенных для создания и (или) обработки графических изображений. В данном классе различают следующие категории: *растровые редакторы*, *векторные редакторы* и программные средства для создания и обработки трехмерной графики (*3D-редакторы*).

Растровые редакторы применяют в тех случаях, когда графический объект представлен в виде комбинации точек, образующих растр и обладающих свойствами яркости и цвета. Такой подход эффективен в тех случаях, когда графическое изображение имеет много полутонов и информация о цвете элементов, составляющих объект, важнее, чем информация об их форме. Это характерно для фотографических и полиграфических изображений. Растровые редакторы широко применяются для обработки изображений, их ретуши, создания фотоэффектов и художественных композиций (коллажей).

Возможности создания новых изображений средствами растровых редакторов ограничены и не всегда удобны. Удобство работы с растровыми редакторами можно значительно повысить, используя специализированные средства ввода информации (графические планшеты). Некоторые художники предпочитают пользоваться традиционными инструментами, после чего вводить рисунок в компьютер с помощью специальных аппаратных средств (*сканеров*) и завершать работу с помощью растрового редактора путем применения спецэффектов.

Векторные редакторы отличаются от растровых способом представления данных об изображении. Элементарным объектом векторного изображения является не точка, а линия. Такой подход характерен для чертежно-графических работ, в которых форма линий имеет большее значение, чем информация о цвете отдельных точек, составляющих ее. В векторных редакторах каждая линия рассматривается как математическая кривая третьего порядка и, соответственно, представляется не комбинацией точек, а математической формулой (в компьютере хранятся числовые коэффициенты этой формулы). Такое представление намного компактнее, чем растровое, соответственно данные занимают много меньше места, однако построение любого объекта выполняется не простым отображением точек на экране, а сопровождается непрерывным пересчетом параметров кривой в координаты экранного или печатного изображения. Соответственно, работа с векторной графикой требует более производительных вычислительных систем.

Из элементарных объектов (линий) создаются простейшие геометрические объекты (примитивы) из которых, в свою очередь, составляются законченные композиции. Художественная иллюстрация, выполненная средствами векторной графики, может содержать десятки тысяч простейших объектов, взаимодействующих друг с другом.

Векторные редакторы удобны для создания изображений, но практически не используются для обработки готовых рисунков. Они нашли широкое применение в рекламном бизнесе, их применяют для оформления обложек полиграфических изданий и всюду, где стиль художественной работы близок к чертежному.

Редакторы трехмерной графики используют для создания трехмерных композиций. Они имеют две характерные особенности. Во-первых, они позволяют

гибко управлять взаимодействием свойств поверхности изображаемых объектов со свойствами источников освещения и, во-вторых, позволяют создавать трехмерную анимацию. Поэтому редакторы трехмерной графики нередко называют также *3D-аниматорами*.

Системы управления базами данных. Базами данных называют огромные массивы данных, организованных в табличные структуры. Основными функциями систем управления базами данных являются:

- ☐ создание пустой (незаполненной) структуры базы данных;
- ☐ предоставление средств ее заполнения или импорта данных из таблиц другой базы;
- ☐ обеспечение возможности доступа к данным, а также предоставление средств поиска и фильтрации.

Многие системы управления базами данных дополнительно предоставляют возможности проведения простейшего анализа данных и их обработки. В результате возможно создание новых таблиц баз данных на основе имеющихся. В связи с широким распространением сетевых технологий к современным системам управления базами данных предъявляется также требование возможности работы с удаленными и распределенными ресурсами, находящимися на серверах всемирной компьютерной сети.

Электронные таблицы. Электронные таблицы предоставляют комплексные средства для хранения различных типов данных и их обработки. В некоторой степени они аналогичны системам управления базами данных, но основной акцент смещен не на хранение массивов данных и обеспечение к ним доступа, а на преобразование данных, причем в соответствии с их внутренним содержанием.

В отличие от баз данных, которые обычно содержат широкий спектр типов данных (от числовых и текстовых до мультимедийных), для электронных таблиц характерна повышенная сосредоточенность на числовых данных. Зато электронные таблицы предоставляют более широкий спектр методов для работы с данными числового типа.

Основное свойство электронных таблиц состоит в том, что при изменении содержания любых ячеек таблицы может происходить автоматическое изменение содержания во всех прочих ячейках, связанных с измененным соотношением, заданным математическими или логическими выражениями (формулами). Простота и удобство работы с электронными таблицами снискали им широкое применение в сфере бухгалтерского учета, в качестве универсальных инструментов анализа финансовых, сырьевых и товарных рынков, доступных средств обработки результатов технических испытаний, то есть всюду, где необходимо автоматизировать регулярно повторяющиеся вычисления достаточно больших объемов числовых данных.

Системы автоматизированного проектирования (САД-системы). Предназначены для автоматизации проектно-конструкторских работ. Применяются в машиностроении, приборостроении, архитектуре. Кроме чертежно-графических работ эти системы позволяют проводить простейшие расчеты (например, расчеты прочности деталей) и выбор готовых конструктивных элементов из обширных баз данных.

Отличительная особенность *CAD*-систем состоит в автоматическом обеспечении на всех этапах проектирования технических условий, норм и правил, что освобождает конструктора (или архитектора) от работ нетворческого характера. Например, в машиностроении *CAD*-системы способны на базе сборочного чертежа изделия автоматически выполнить рабочие чертежи деталей, подготовить необходимую технологическую документацию с указанием последовательности переходов механической обработки, назначить необходимые инструменты, станочные и контрольные приспособления, а также подготовить управляющие программы для станков с числовым программным управлением (ЧПУ), промышленных роботов и гибких автоматизированных линий. Сегодня системы автоматизированного проектирования являются необходимым компонентом, без которого теряется эффективность реализации гибких производственных систем (ГПС) и автоматизированных систем управления технологическими процессами (АСУТП).

Настольные издательские системы. Назначение программ этого класса состоит в автоматизации процесса верстки полиграфических изданий. Этот класс программного обеспечения занимает промежуточное положение между текстовыми процессорами и системами автоматизированного проектирования.

Теоретически текстовые процессоры предоставляют средства для внедрения в текстовый документ объектов другой природы, например объектов векторной и растровой графики, а также позволяют управлять взаимодействием между параметрами текста и параметрами внедренных объектов. Однако на практике для изготовления полиграфической продукции эти средства либо функционально недостаточны с точки зрения требований полиграфии, либо недостаточно удобны для производительной работы.

От текстовых процессоров настольные издательские системы отличаются расширенными средствами управления взаимодействием текста с параметрами страницы и с графическими объектами. С другой стороны, они отличаются пониженными функциональными возможностями по автоматизации ввода и редактирования текста. Типичный прием использования настольных издательских систем состоит в том, что их применяют к документам, прошедшим предварительную обработку в текстовых процессорах и графических редакторах.

Экспертные системы. Предназначены для анализа данных, содержащихся в *базах знаний*, и выдачи рекомендаций по запросу пользователя. Такие системы применяют в тех случаях, когда исходные данные хорошо формализуются, но для принятия решения требуются обширные специальные знания. Характерными областями использования экспертных систем являются юриспруденция, медицина, фармакология, химия. По совокупности признаков заболевания медицинские экспертные системы помогают установить диагноз и назначить лекарства, дозировку и программу лечебного курса. По совокупности признаков события юридические экспертные системы могут дать правовую оценку и предложить порядок действий как для стороны обвинения, так и для стороны защиты.

Характерной особенностью экспертных систем является их способность к *саморазвитию*. Исходные данные хранятся в базе знаний в виде *фактов*, между которыми с помощью специалистов-экспертов устанавливается определенная система *отношений*. Если на этапе тестирования экспертной системы устанавливается, что

она дает некорректные рекомендации и заключения по конкретным вопросам или не может дать их вообще, это означает либо отсутствие важных фактов в ее базе, либо нарушения в логической системе отношений. И том и в другом случае экспертная система сама может сгенерировать достаточный набор запросов к эксперту и автоматически повысить свое качество.

С использованием экспертных систем связана особая область научно-технической деятельности, называемая *инженерией знаний*. Инженеры знаний — это специалисты особой квалификации, выступающие в качестве промежуточного звена между разработчиками экспертной системы (программистами) и ведущими специалистами в конкретных областях науки и техники (экспертами).

Веб-редакторы. Это особый класс редакторов, объединяющих в себе свойства текстовых и графических редакторов. Они предназначены для создания и редактирования так называемых *веб-документов (веб-страниц Интернета)*. Веб-документы — это электронные документы, при подготовке которых следует учитывать ряд особенностей, связанных с приемом/передачей информации в Интернете.

Теоретически для создания веб-документов можно использовать обычные текстовые редакторы и процессоры, а также некоторые из графических редакторов векторной графики, но веб-редакторы обладают рядом полезных функций, повышающих производительность труда веб-дизайнеров. Программы этого класса можно также эффективно использовать для подготовки электронных документов и мультимедийных изданий.

Браузеры (обозреватели, средства просмотра Web). К этой категории относятся программные средства, предназначенные для просмотра электронных документов, выполненных в формате HTML (документы этого формата используются в качестве веб-документов). Современные браузеры воспроизводят не только текст и графику. Они могут воспроизводить музыку, человеческую речь, обеспечивать прослушивание радиопередач в Интернете, просмотр видеоконференций, работу со службами электронной почты, с системой телеконференций (групп новостей) и многое другое.

Интегрированные системы делопроизводства. Представляют собой программные средства автоматизации рабочего места руководителя. К основным функциям подобных систем относятся функции создания, редактирования и форматирования простейших документов, централизация функций электронной почты, факсимильной и телефонной связи, диспетчеризация и мониторинг документооборота предприятия, координация деятельности подразделений, оптимизация административно-хозяйственной деятельности и поставка по запросу оперативной и справочной информации.

Бухгалтерские системы. Это специализированные системы, сочетающие в себе функции текстовых и табличных редакторов, электронных таблиц и систем управления базами данных. Предназначены для автоматизации подготовки первичных бухгалтерских документов предприятия и их учета, для ведения счетов плана бухгалтерского учета, а также для автоматической подготовки регулярных отчетов по итогам производственной, хозяйственной и финансовой деятельности в форме, принятой для предоставления в налоговые органы, внебюджетные фонды и органы статистического учета. Несмотря на то что теоретически все функции, характерные

для бухгалтерских систем, можно исполнять и другими вышеперечисленными программными средствами, использование бухгалтерских систем удобно благодаря интеграции разных средств в одной системе.

При решении о внедрении на предприятии автоматизированной системы бухгалтерского учета необходимо учитывать необходимость наличия в ней средств адаптации при изменении нормативно-правовой базы. В связи с тем, что в данной области нормативно-правовая база в России отличается крайней нестабильностью и подвержена частым изменениям, возможность гибкой перенастройки системы является обязательной функцией, хотя это требует от пользователей системы повышенной квалификации.

Финансовые аналитические системы. Программы этого класса используются в банковских и биржевых структурах. Они позволяют контролировать и прогнозировать ситуацию на финансовых, товарных и сырьевых рынках, производить анализ текущих событий, готовить сводки и отчеты.

Геоинформационные системы (ГИС). Предназначены для автоматизации картографических и геодезических работ на основе информации, полученной топографическими или аэрокосмическими методами.

Системы видеомонтажа. Предназначены для цифровой обработки видеоматериалов, их монтажа, создания видеоэффектов, устранения дефектов, наложения звука, титров и субтитров.

Отдельные категории прикладных программных средств, обладающие своими развитыми внутренними системами классификации, представляют *обучающие, развивающие, справочные и развлекательные* системы и программы. Характерной особенностью этих классов программного обеспечения являются повышенные требования к мультимедийной составляющей (использование музыкальных композиций, средств графической анимации и видеоматериалов).

Классификация служебных программных средств

Диспетчеры файлов (файловые менеджеры). С помощью программ данного класса выполняется большинство операций, связанных с обслуживанием файловой структуры: копирование, перемещение и переименование файлов, создание каталогов (папок), удаление файлов и каталогов, поиск файлов и навигация в файловой структуре. Базовые программные средства, предназначенные для этой цели, обычно входят в состав программ системного уровня и устанавливаются вместе с операционной системой. Однако для повышения удобства работы с компьютером большинство пользователей устанавливают дополнительные служебные программы.

Средства сжатия данных (архиваторы). Предназначены для создания архивов. Архивирование данных упрощает их хранение за счет того, что большие группы файлов и каталогов сводятся в один архивный файл. При этом повышается и эффективность использования носителя за счет того, что архивные файлы обычно имеют повышенную плотность записи информации. Архиваторы часто используют для создания резервных копий ценных данных.

Средства просмотра и воспроизведения. Обычно для работы с файлами данных необходимо загрузить их в «родительскую» прикладную систему, с помощью

которой они были созданы. Это дает возможность просматривать документы и вносить в них изменения. Но в тех случаях, когда требуется только просмотр без редактирования, удобно использовать более простые и более универсальные средства, позволяющие просматривать документы разных типов.

В тех случаях, когда речь идет о звукозаписи или видеозаписи, вместо термина *просмотр* применяют термин *воспроизведение* документов.

Средства диагностики. Предназначены для автоматизации процессов диагностики программного и аппаратного обеспечения. Они выполняют необходимые проверки и выдают собранную информацию в удобном и наглядном виде. Их используют не только для устранения неполадок, но и для оптимизации работы компьютерной системы.

Средства контроля (мониторинга). Программные средства контроля иногда называют *мониторами*. Они позволяют следить за процессами, происходящими в компьютерной системе. При этом возможны два подхода: наблюдение в реальном режиме времени или контроль с записью результатов в специальном протокольном файле. Первый подход обычно используют при изыскании путей для оптимизации работы вычислительной системы и повышения ее эффективности. Второй подход используют в тех случаях, когда мониторинг выполняется автоматически и (или) дистанционно. В последнем случае результаты мониторинга можно передать удаленной службе технической поддержки для установления причин конфликтов в работе программного и аппаратного обеспечения.

Средства мониторинга, работающие в режиме реального времени, особенно полезны для практического изучения приемов работы с компьютером, поскольку позволяют наглядно отображать те процессы, которые обычно скрыты от глаз пользователя.

Мониторы установки. Программы этой категории предназначены для контроля над установкой программного обеспечения. Необходимость в данном программном обеспечении связана с тем, что между различными категориями программного обеспечения могут устанавливаться связи. Вертикальные связи (между уровнями) являются необходимым условием функционирования всех компьютеров. Горизонтальные связи (внутри уровней) характерны для компьютеров, работающих с операционными системами, поддерживающими принцип совместного использования одних и тех же ресурсов разными программными средствами. И в тех и в других случаях при установке или удалении программного обеспечения могут происходить нарушения работоспособности прочих программ.

Мониторы установки следят за состоянием и изменением окружающей программной среды, отслеживают и протоколируют образование новых связей и позволяют восстанавливать связи, утраченные в результате удаления ранее установленных программ.

Простейшие средства управления установкой и удалением программ обычно входят в состав операционной системы и размещаются на системном уровне программного обеспечения, однако они редко бывают достаточны. Поэтому в вычис-

лительных системах, требующих повышенной надежности, используют дополнительные служебные программы.

Средства коммуникации (коммуникационные программы). С появлением электронной связи и компьютерных сетей программы этого класса приобрели очень большое значение. Они позволяют устанавливать соединения с удаленными компьютерами, обслуживают передачу сообщений электронной почты, работу с телеконференциями (группами новостей), обеспечивают пересылку факсимильных сообщений и выполняют множество других операций в компьютерных сетях.

Средства обеспечения компьютерной безопасности. К этой весьма широкой категории относятся средства пассивной и активной защиты данных от повреждения, а также средства защиты от несанкционированного доступа, просмотра и изменения данных.

В качестве *средств пассивной защиты* используют служебные программы, предназначенные для резервного копирования. Нередко они обладают и базовыми свойствами диспетчеров архивов (архиваторов). В качестве *средств активной защиты* применяют *антивирусное программное обеспечение*. Для защиты данных от несанкционированного доступа, их просмотра и изменения служат специальные системы, основанные на криптографии.

Понятие об информационном и математическом обеспечении вычислительных систем

Наряду с аппаратным и программным обеспечением средств вычислительной техники в некоторых случаях целесообразно рассматривать *информационное обеспечение*, под которым понимают совокупность программ и предварительно подготовленных данных, необходимых для работы данных программ.

Рассмотрим, например, систему автоматической проверки орфографии в редактируемом тексте. Ее работа заключается в том, что лексические единицы исходного текста сравниваются с заранее заготовленным эталонным массивом данных (словарем). В данном случае для успешной работы системы необходимо иметь кроме аппаратного и программного обеспечения специальные наборы словарей, подключаемые извне. Это пример *информационного обеспечения* вычислительной техники.

В специализированных компьютерных системах (бортовых компьютерах автомобилей, судов, ракет, самолетов, космических летательных аппаратов и т. п.) совокупность программного и информационного обеспечения называют *математическим обеспечением*. Как правило, оно «жестко» записывается в микросхемы ПЗУ и может быть изменено только путем замены ПЗУ или его перепрограммирования на специальном оборудовании.

Подведение итогов

Вычислительная техника прошла те же исторические этапы эволюции, которые прошли и все прочие технические устройства: от ручных приспособлений к механическим устройствам и далее к гибким автоматическим системам. Современный компьютер — это прибор. Его принцип действия — электронный, а назначе-

ние — автоматизация операций с данными. Гибкость автоматизации основана на том, что операции с данными выполняются по заранее заготовленным и легко сменяемым программам. Универсальность компьютеров основана на том, что любые типы данных представляются в нем с помощью универсального двоичного кодирования.

Работа компьютерной системы протекает в непрерывном взаимодействии аппаратных и программных средств. Физически аппаратные средства согласуются друг с другом с помощью механических и электрических разъемов и контактов. Логически они согласуются друг с другом с помощью программ, называемых *драйверами устройств*.

Работа компьютерных программ имеет многоуровневый характер. Программы низшего (базового) уровня занимаются только взаимодействием с базовыми аппаратными средствами и согласованием их работы. Ключевая роль программ базового уровня проявляется в момент первичного запуска компьютера.

Программы *системного уровня* опираются на программы базового уровня и обеспечивают взаимодействие пользователя с оборудованием, взаимодействие дополнительного оборудования с базовым, а также предоставляют возможность для установки и работы программ более высоких уровней.

Программы *служебного уровня* выполняют обслуживание компьютерной системы, обеспечивают ее контроль и настройку. В своей работе они опираются на программы базового и системного уровней.

Программы *прикладного уровня* используются человеком для исполнения практических задач с помощью компьютера. Эти программы опираются на программы нижележащих уровней.

Совокупность программ, установленных на компьютере, называется его *программной конфигурацией*. Совокупность оборудования, подключенного к компьютеру, называется его *аппаратной конфигурацией*. Несмотря на то что по своей архитектуре и функциональному назначению разные компьютеры могут быть весьма близки друг другу, найти два компьютера, имеющих одинаковые аппаратные и программные конфигурации, практически невозможно. На каждом рабочем месте программно-аппаратная конфигурация создается такой, чтобы наиболее эффективно решать конкретные практические задачи, характерные для данного рабочего места.

Вопросы для самоконтроля

1. В чем вы видите диалектический характер связи между *программным обеспечением* и *аппаратным*?
2. Назовите четыре основных уровня программного обеспечения. Каков порядок их взаимодействия?
3. К какому классу относятся программные средства, встроенные в видеомаягнитофон, программируемую стиральную машину, СВЧ-плиту?
4. В чем преимущества и недостатки выполнения офисных работ (например, копирования-множительных) аппаратными и программными средствами?

5. Какие категории программного обеспечения могут быть использованы в работе малого предприятия и для каких целей?
6. Какие виды работ, характерные для крупного промышленного предприятия (например, машиностроительного завода), могут быть автоматизированы с помощью компьютеров? Какие категории программных средств для этого необходимы?
7. Назовите основные категории программного обеспечения, относящиеся к классу графических редакторов. В чем состоит принципиальная разница между этими категориями?
8. Что общего и в чем различие между понятиями *программное обеспечение* и *информационное обеспечение* средств вычислительной техники?

Глава третья

Устройство персонального компьютера

3.1. Базовая аппаратная конфигурация персонального компьютера

Персональный компьютер — универсальная техническая система. Его *конфигурацию* (состав оборудования) можно гибко изменять по мере необходимости. Тем не менее существует понятие *базовой конфигурации*, которую считают типовой. В таком комплекте компьютер обычно поставляется. Понятие базовой конфигурации может меняться. В настоящее время в базовой конфигурации рассматривают четыре устройства (рис. 3.1):

- ☐ системный блок;
- ☐ монитор;
- ☐ клавиатура;
- ☐ мышь.

Системный блок

Системный блок представляет собой основной узел, внутри которого установлены наиболее важные компоненты. Устройства, находящиеся внутри системного блока, называют *внутренними*, а устройства, подключаемые к нему снаружи, — *внешними*. Внешние дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных, также называют *периферийными*.

По внешнему виду системные блоки различаются формой корпуса. Корпуса персональных компьютеров выпускают в горизонтальном (desktop) и вертикальном (tower) исполнении. Корпуса, имеющие вертикальное исполнение, различают по габаритам: *полноразмерный* (big tower), *среднеразмерный* (midi tower) и *малоразмерный* (mini tower). Среди корпусов, имеющих горизонтальное исполнение, выделяют *плоские* и *особо плоские* (slim).

Кроме формы, для корпуса важен параметр, называемый *форм-фактором*. От него зависят требования к размещаемым устройствам. Прежним стандартом корпуса персональных компьютеров был форм-фактор AT, в настоящее время в основном используются корпуса форм-фактора ATX. Форм-фактор корпуса должен быть обязательно согласован с форм-фактором главной (системной) платы компьютера, так называемой *материнской платы* (см. ниже).

Корпуса персональных компьютеров поставляются вместе с блоком питания, и, таким образом, мощность блока питания также является одним из параметров корпуса. Для массовых моделей достаточной является мощность блока питания 250–300 Вт.



Рис. 3.1. Базовая конфигурация компьютерной системы

Монитор

Монитор — устройство визуального представления данных. Это не единственно возможное, но главное устройство вывода. Его основными потребительскими параметрами являются: тип, размер и шаг маски экрана, максимальная частота регенерации изображения, класс защиты.

Сейчас наиболее распространены плоские жидкокристаллические мониторы (ЖК). Они пришли на смену мониторам, основанным на электронно-лучевой трубке (ЭЛТ). Электронно-лучевые мониторы все еще используются, однако их становится все меньше. Ранние модели ЖК-мониторов уступали ЭЛТ-устройствам в качестве изображения и в некоторых других характеристиках, однако в наши дни практически все проблемы решены.

Размер монитора измеряется между противоположными углами видимой части экрана по диагонали. Единица измерения — дюймы. Стандартные размеры: 14"; 15"; 17"; 19"; 20"; 21". В настоящее время наиболее универсальными являются мониторы размером 21 (ЖК) и 19 дюймов (ЭЛТ).

Изображение на экране ЭЛТ-монитора получается в результате облучения люминофорного покрытия остронаправленным пучком электронов, разогнанных в вакуумной колбе. Для получения цветного изображения люминофорное покрытие имеет точки или полосы трех типов, светящиеся красным, зеленым и синим цветом. Чтобы на экране все три луча сходились строго в одну точку и изображение было четким, перед люминофором ставят маску — панель с регулярно расположенными отверстиями или щелями. Часть мониторов оснащена маской из вертикальных проволочек, что усиливает яркость и насыщенность изображения.

Чем меньше шаг между отверстиями или щелями (*шаг маски*), тем четче и точнее полученное изображение. Шаг маски измеряют в долях миллиметра. В настоящее время наиболее распространены мониторы с шагом маски 0,24–0,26 мм. Устаревшие мониторы могут иметь шаг до 0,43 мм, что негативно сказывается на органах зрения при работе с компьютером. Модели повышенной стоимости могут иметь значение менее 0,24 мм.

На экране жидкокристаллического монитора изображение образуется в результате прохождения белого света лампы подсветки через ячейки, прозрачность которых зависит от приложенного напряжения. Элементарная триада состоит из трех ячеек зеленого, красного и синего цвета и соответствует одному пикселу экрана. Размер монитора по диагонали и разрешение экрана однозначно определяют размер такой триады и тем самым зернистость изображения.

Частота регенерации (обновления) изображения показывает, сколько раз в течение секунды монитор может полностью сменить изображение (поэтому ее также называют *частотой кадров*). Этот параметр зависит не только от монитора, но и от свойств и настроек видеоадаптера (см. ниже), хотя предельные возможности определяет все-таки монитор.

Частоту регенерации изображения измеряют в герцах (Гц). Чем она выше, тем четче и устойчивее изображение, тем меньше утомление глаз, тем больше времени можно работать с компьютером непрерывно. При частоте регенерации порядка 60 Гц мелкое мерцание изображения может быть заметно невооруженным глазом. Сегодня такое значение считается недопустимым. Для ЭЛТ-мониторов минимальным считают значение 75 Гц, нормативным — 85 Гц и комфортным — 100 Гц и более. У жидкокристаллических мониторов изображение более инерционно, так что мерцание подавляется автоматически. Для них частота обновления в 75 Гц уже считается комфортной.

Класс защиты монитора определяется стандартом, которому соответствует монитор с точки зрения требований техники безопасности. В настоящее время общепризнанными считаются следующие международные стандарты: MPR-II, TCO-92, TCO-95, TCO-99, TCO-03, TCO-06 (приведены в хронологическом порядке). Стандарт MPR-II ограничил уровни электромагнитного излучения пределами, безопасными для человека. В стандарте TCO-92 эти нормы были сохранены, а в стандартах TCO-95 и TCO-99 — ужесточены. Эргономические и экологические нормы впервые появились в стандарте TCO-95, а стандарт TCO-99 установил самые жесткие нормы по параметрам, определяющим качество изображения (яркость, контрастность, мерцание, антибликовые свойства покрытия). Стандарт TCO-03 посвящен ЭЛТ и ЖК-дисплеям, именно по этому стандарту сертифицируются большинство современных офисных мониторов. Стандарт TCO-06 содержит набор требований, по которым должны сертифицироваться так называемые медиадисплеи — дисплеи, одной из основных функций которых является вывод мультимедийных изображений.

Большинством параметров изображения, полученного на экране монитора, можно управлять программно. Программные средства, предназначенные для этой цели, обычно входят в системный комплект программного обеспечения — мы рассмотрим их при изучении операционной системы компьютера.

Клавиатура

Клавиатура — клавишное устройство управления персональным компьютером. Служит для ввода *алфавитно-цифровых (знаковых)* данных, а также команд управления. Комбинация монитора и клавиатуры обеспечивает простейший *интерфейс пользователя*. С помощью клавиатуры управляют компьютерной системой, а с помощью монитора получают от нее отклик.

Принцип действия. Клавиатура относится к стандартным средствам персонального компьютера. Ее основные функции не нуждаются в поддержке специальными системными программами (драйверами). Необходимое программное обеспечение для начала работы с компьютером уже имеется в микросхеме ПЗУ в составе базовой системы ввода-вывода (BIOS), и потому компьютер реагирует на нажатия клавиш сразу после включения.

Принцип действия клавиатуры заключается в следующем.

1. При нажатии на клавишу (или комбинацию клавиш) специальная микросхема, встроенная в клавиатуру, генерирует и выдает так называемый *скан-код*.
2. Скан-код поступает в микросхему, выполняющую функции *порта* клавиатуры. (Порты — специальные аппаратно-логические устройства, отвечающие за связь процессора с другими устройствами.) Порт клавиатуры — это довольно простое устройство, интегрированное в одну из микросхем материнской платы.
3. Порт клавиатуры выдает процессору прерывание с фиксированным номером. Для клавиатуры номер прерывания — 9 (Interrupt 9, Int 9).
4. Получив прерывание, процессор откладывает текущую работу и по номеру прерывания обращается в специальную область оперативной памяти, в которой находится так называемый *вектор прерываний*. Вектор прерываний — это список адресных данных с фиксированной длиной записи. Каждая запись содержит адрес программы, которая должна обслужить прерывание с номером, совпадающим с номером записи.
5. Определив адрес начала программы, обрабатывающей возникшее прерывание, процессор переходит к ее исполнению. Простейшая программа обработки клавиатурного прерывания «зашиита» в микросхему ПЗУ, но программисты могут «подставить» вместо нее свою программу, если изменят данные в векторе прерываний.
6. Программа-обработчик прерывания направляет процессор к порту клавиатуры, где он находит скан-код, загружает его в свои регистры, потом под управлением обработчика определяет, какой код символа соответствует данному скан-коду.
7. Далее обработчик прерываний отправляет полученный код символа в небольшую область памяти, известную как *буфер клавиатуры*, и прекращает свою работу, известив об этом процессор.
8. Процессор прекращает обработку прерывания и возвращается к отложенной задаче.
9. Введенный символ хранится в буфере клавиатуры до тех пор, пока его не заберет оттуда та программа, для которой он предназначался, например текстовый редактор или текстовый процессор. Если символы поступают в буфер чаще, чем

забираются оттуда, возможен эффект переполнения буфера. В этом случае ввод новых символов на некоторое время прекращается. На практике в этот момент при нажатии на клавишу мы слышим предупреждающий звуковой сигнал и не наблюдаем ввода данных.



Рис. 3.2. Группы клавиш стандартной клавиатуры

Состав клавиатуры. Стандартная клавиатура имеет более 100 клавиш, функционально распределенных по нескольким группам (см. рис. 3.2).

Группа *алфавитно-цифровых клавиш* предназначена для ввода знаковой информации и команд, набираемых по буквам. Каждая клавиша может работать в нескольких режимах (*регистрах*) и, соответственно, может использоваться для ввода нескольких символов. Переключение между *нижним регистром* (для ввода строчных символов) и *верхним регистром* (для ввода прописных символов) выполняют удержанием клавиши **SHIFT** (нефиксированное переключение). При необходимости жестко переключить регистр используют клавишу **CAPS LOCK** (фиксированное переключение). Если клавиатура используется для ввода данных, абзац закрывают нажатием клавиши **ENTER**. При этом автоматически начинается ввод текста с новой строки. Если клавиатуру используют для ввода команд, клавишей **ENTER** завершают ввод команды и начинают ее исполнение.

Для разных языков существуют различные схемы закрепления символов национальных алфавитов за конкретными алфавитно-цифровыми клавишами. Такие схемы называются *раскладками клавиатуры*. Переключения между различными раскладками выполняются программным образом — это одна из функций операционной системы. Соответственно, способ переключения зависит от того, в какой операционной системе работает компьютер. Например, в системе *Windows 7* для этой цели могут использоваться следующие комбинации: левая клавиша **ALT+SHIFT** или **CTRL+SHIFT**. При работе с другой операционной системой способ переключения можно установить по справочной системе той программы, которая выполняет переключение.

Общепринятые раскладки клавиатуры имеют свои корни в раскладках клавиатур пишущих машинок. Для персональных компьютеров *IBM PC* типовыми считаются раскладки **QWERTY** (английская) и **ЙЦУКЕН** (русская). Раскладки принято именовать по символам, закрепленным за первыми клавишами верхней строки алфавитной группы.

Группа функциональных клавиш включает двенадцать клавиш (от F1 до F12), размещенных в верхней части клавиатуры. Функции, закрепленные за данными клавишами, зависят от свойств конкретной работающей в данный момент программы, а в некоторых случаях и от свойств операционной системы. Общепринятым для большинства программ является соглашение о том, что клавиша F1 вызывает справочную систему, в которой можно найти справку о действии прочих клавиш.

Служебные клавиши располагаются рядом с клавишами алфавитно-цифровой группы. В связи с тем, что ими приходится пользоваться особенно часто, они имеют увеличенный размер. К ним относятся рассмотренные выше клавиши **SHIFT** и **ENTER**, регистровые клавиши **ALT** и **CTRL** (их используют в комбинации с другими клавишами для формирования команд), клавиша **TAB** (для ввода позиций табуляции при наборе текста), клавиша **ESC** (от английского слова *escape*) для отказа от исполнения начатой операции и клавиша **BACKSPACE** для удаления только что введенных знаков (она находится над клавишей **ENTER** и часто маркируется стрелкой, направленной влево).

Служебные клавиши **PRINT SCREEN**, **SCROLL LOCK** и **PAUSE/BREAK** размещаются справа от группы функциональных клавиш и выполняют специфические функции, зависящие от действующей операционной системы. Общепринятыми являются следующие действия:

- **PRINT SCREEN** — сохранение текущего состояния экрана в специальной области оперативной памяти, называемой *буфером обмена*.
- **SCROLL LOCK** — переключение режима работы в некоторых (как правило, устаревших) программах.
- **PAUSE/BREAK** — приостановка/прерывание текущего процесса (обычно используется в комбинации с клавишей **CTRL**, доступно лишь в некоторых программах).

Две группы *клавиш управления курсором* расположены справа от алфавитно-цифровой панели. *Курсором* называется экранный элемент, указывающий место ввода знаковой информации. Курсор используется при работе с программами, выполняющими ввод данных и команд с клавиатуры. Клавиши управления курсором позволяют управлять позицией ввода.

Четыре клавиши со стрелками выполняют смещение курсора в направлении, указанном стрелкой (их обычно называют просто *курсорными клавишами*). Действие прочих клавиш описано ниже.

PAGE UP/PAGE DOWN — перевод курсора на одну страницу вверх или вниз. Понятие «страница» обычно относится к фрагменту документа, видимому на экране. В графических операционных системах (например, Windows) этими клавишами выполняют «прокрутку» содержимого в текущем окне. Действие этих клавиш во многих программах может быть модифицировано с помощью служебных регистро-

вых клавиш, в первую очередь **SHIFT** и **CTRL**. Конкретный результат модификации зависит от конкретной программы и/или операционной системы.

Клавиши **HOME** и **END** переводят курсор в начало или конец текущей строки соответственно. Их действие также модифицируется регистровыми клавишами.

Традиционное назначение клавиши **INSERT** состоит в переключении режима ввода данных (переключение между режимами *вставки* и *замены*). Если текстовый курсор находится внутри существующего текста, то в режиме вставки происходит ввод новых знаков без замены существующих символов (текст как бы раздвигается). В режиме замены новые знаки заменяют текст, имевшийся ранее в позиции ввода.

В современных программах действие клавиши **INSERT** может быть иным. Конкретную информацию следует получить в справочной системе программы. Возможно, что действие этой клавиши является настраиваемым, — это также зависит от свойств конкретной программы.

Клавиша **DELETE** предназначена для удаления знаков, находящихся справа от текущего положения курсора. При этом положение позиции ввода остается неизменным.

Сравните действие клавиши **DELETE** с действием служебной клавиши **BACKSPACE**. Последняя служит для удаления знаков, но при ее использовании позиция ввода смещается влево, и, соответственно, удаляются символы, находящиеся не справа, а слева от курсора.

Группа клавиш дополнительной панели дублирует действие цифровых и некоторых знаковых клавиш основной панели. Во многих случаях для использования этой группы клавиш следует предварительно включать клавишу-переключатель **NUM LOCK** (о состоянии переключателей **NUM LOCK**, **CAPS LOCK** и **SCROLL LOCK** можно судить по светодиодным индикаторам, обычно расположенным в правом верхнем углу клавиатуры).

Появление дополнительной панели клавиатуры относится к началу 80-х годов. В то время клавиатуры были относительно дорогостоящими устройствами. Первоначальное назначение дополнительной панели состояло в снижении износа основной панели при проведении расчетно-кассовых вычислений, а также при управлении компьютерными играми (при выключенном переключателе **NUM LOCK** клавиши дополнительной панели могут использоваться в качестве клавиш управления курсором).

В наши дни клавиатуры относят к малоценным быстроизнашивающимся устройствам и приспособлениям, и существенной необходимости оберегать их от износа нет. Тем не менее за дополнительной клавиатурой сохраняется важная функция ввода символов, для которых известен расширенный код ASCII (см. выше), но неизвестно закрепление за клавишей клавиатуры. Так, например, известно, что символ «§» (параграф) имеет код 0167, а символ «°» (угловой градус) имеет код 0176, но соответствующих им клавиш на клавиатуре нет. В таких случаях для их ввода используют дополнительную панель.

Порядок ввода символов по известному ALT-коду.

1. Убедиться в том, что включен переключатель **NUM LOCK**.
2. Нажать и удерживать клавишу **ALT**.

3. Не отпуская клавиши ALT, набрать последовательно на дополнительной панели ALT-код вводимого символа, например 0167.
4. Отпустить клавишу ALT. Символ, имеющий код 0167, появится на экране в позиции ввода.

Узнать ALT-коды некоторых символов позволяет программа Таблица символов (см. раздел 7.3).

Настройка клавиатуры. Клавиатуры персональных компьютеров обладают *свойством повтора знаков*, которое используется для автоматизации процесса ввода. Оно состоит в том, что при длительном удержании клавиши начинается автоматический ввод связанного с ней кода. При этом настраиваемыми параметрами являются:

- ☐ интервал времени после нажатия, по истечении которого начнется автоматический повтор кода;
- ☐ темп повтора (количество знаков в секунду).

Средства настройки клавиатуры относятся к системным и обычно входят в состав операционной системы. Кроме параметров режима повтора, настройке подлежат также используемые раскладки и органы управления, используемые для переключения раскладок. Со средствами настройки клавиатуры мы познакомимся при изучении функций операционной системы.

Мышь

Мышь — устройство управления манипуляторного типа. Представляет собой плоскую коробочку с двумя-тремя кнопками. Перемещение мыши по плоской поверхности синхронизировано с перемещением графического объекта (*указателя мыши*) на экране монитора.

Принцип действия. В отличие от рассмотренной ранее клавиатуры мышь не является стандартным органом управления, и персональный компьютер не имеет для нее выделенного порта. Для мыши нет и постоянного выделенного прерывания, а базовые средства ввода и вывода (BIOS) компьютера, размещенные в постоянном запоминающем устройстве (ПЗУ), не содержат программных средств для обработки прерываний мыши.

В связи с этим в первый момент после включения компьютера мышь не работает. Она нуждается в поддержке специальной системной программы — *драйвера мыши*. Драйвер устанавливается либо при первом подключении мыши, либо при установке операционной системы компьютера. Хотя мышь и не имеет выделенного порта на материнской плате, для работы с ней используют один из стандартных портов, средства для работы с которыми имеются в составе BIOS. Драйвер мыши предназначен для интерпретации сигналов, поступающих через порт. Кроме того, он обеспечивает механизм передачи информации о положении и состоянии мыши операционной системе и работающим программам.

Компьютером управляют перемещением мыши по плоскости и кратковременными нажатиями правой и левой кнопок. (Эти нажатия называются *щелчками*.)

В отличие от клавиатуры мышь не может напрямую использоваться для ввода знаковой информации — ее принцип управления является *событийным*. Перемещения мыши и щелчки ее кнопок являются *событиями* с точки зрения ее программы-драйвера. Анализируя эти события, драйвер устанавливает, когда произошло событие и в каком месте экрана в этот момент находился указатель. Эти данные передаются в прикладную программу, с которой работает пользователь в данный момент. По ним программа может определить команду, которую имел в виду пользователь, и приступить к ее исполнению.

Комбинация монитора и мыши обеспечивает наиболее современный тип интерфейса пользователя, который называется *графическим*. Пользователь наблюдает на экране графические *объекты* и *элементы управления*. С помощью мыши он изменяет *свойства объектов* и приводит в действие *элементы управления* компьютерной системой, а с помощью монитора получает от нее отклик в графическом виде.

Стандартная мышь имеет только две кнопки, хотя существуют нестандартные мыши с тремя кнопками. Сегодня наиболее распространены мыши, в которых роль третьей кнопки играет вращающееся колесико-регулятор. Функции дополнительных органов управления определяются тем программным обеспечением, которое поставляется вместе с устройством.

К числу регулируемых параметров мыши относятся: *чувствительность* (выражает величину перемещения указателя на экране при заданном линейном перемещении мыши), функции левой и правой кнопок, а также *чувствительность к двойному нажатию* (максимальный интервал времени, при котором два щелчка кнопкой мыши расцениваются как один двойной щелчок). Программные средства, предназначенные для этих регулировок, обычно входят в системный комплект программного обеспечения — мы рассмотрим их при изучении операционной системы.

3.2. Внутренние устройства системного блока

Материнская плата

Материнская плата — основная плата персонального компьютера. На ней размещаются:

- ❑ *процессор* — основная микросхема, выполняющая большинство математических и логических операций;
- ❑ *микропроцессорный комплект* (чипсет) — набор микросхем, управляющих работой внутренних устройств компьютера и определяющих основные функциональные возможности материнской платы;
- ❑ *шины* — наборы проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера;
- ❑ *оперативная память* (оперативное запоминающее устройство, ОЗУ) — набор микросхем, предназначенных для временного хранения данных, когда компьютер включен;

- ПЗУ (постоянное запоминающее устройство) — микросхема, предназначенная для длительного хранения данных, в том числе и когда компьютер выключен;
- разъемы для подключения дополнительных устройств (*слоты*).

Устройства, входящие в состав материнской платы, рассматриваются отдельно в разделе 3.3.

Жесткий диск

Жесткий диск — основное устройство для долговременного хранения больших объемов данных и программ. На самом деле это не один диск, а группа соосных дисков, имеющих магнитное покрытие и вращающихся с высокой скоростью. Таким образом, этот «диск» имеет не две поверхности, как должно быть у обычного плоского диска, а $2n$ поверхностей, где n — число отдельных дисков в группе.

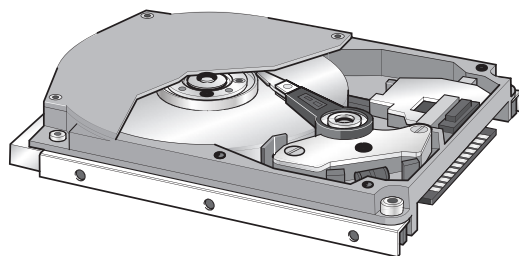


Рис. 3.3. Жесткий диск

Над каждой поверхностью располагается головка, предназначенная для чтения-записи данных. При высоких скоростях вращения дисков (90–250 об/с) в зазоре между головкой и поверхностью образуется аэродинамическая подушка, и головка парит над магнитной поверхностью на высоте, составляющей несколько тысячных долей миллиметра. При изменении силы тока, протекающего через головку, происходит изменение напряженности динамического магнитного поля в зазоре, что вызывает изменения в стационарном магнитном поле ферромагнитных частиц, образующих покрытие диска. Так осуществляется запись данных на магнитный диск.

Операция считывания происходит в обратном порядке. Намагниченные частицы покрытия, проносящиеся на высокой скорости вблизи головки, наводят в ней ЭДС самоиндукции. Электромагнитные сигналы, возникающие при этом, усиливаются и передаются на обработку.

Управление работой жесткого диска выполняет специальное аппаратно-логическое устройство — *контроллер жесткого диска*. В прошлом оно представляло собой отдельную *дочернюю плату*, которую подключали к одному из свободных слотов материнской платы. В настоящее время функции контроллеров дисков частично интегрированы в сам жесткий диск, а частично выполняются микросхемами, входящими в микропроцессорный комплект (чипсет), хотя некоторые виды высокопроизводительных контроллеров жестких дисков по-прежнему могут поставляться на отдельной плате.

К основным параметрам жестких дисков относятся *емкость* и *производительность*. Емкость дисков зависит от технологии их изготовления. В настоящее время большинство производителей жестких дисков используют изобретенную компанией IBM технологию с использованием *гигантского магниторезистивного эффекта* (GMR — Giant Magnetic Resistance). В настоящее время на пластину может приходиться 500 и более гигабайт, но развитие продолжается.

С другой стороны, производительность жестких дисков меньше зависит от технологии их изготовления. Сегодня все жесткие диски имеют очень высокий показатель скорости внутренней передачи данных (до 40–100 Мбайт/с), и потому их производительность в первую очередь зависит от характеристик интерфейса, с помощью которого они связаны с материнской платой. В зависимости от типа интерфейса разброс значений может быть очень большим: от нескольких мегабайт в секунду до 13–16 Мбайт/с для интерфейсов типа EIDE; до 80 Мбайт/с для интерфейсов типа SCSI и от 50 Мбайт/с и более для наиболее современных интерфейсов типа IEEE 1394 и Serial ATA.

Кроме скорости передачи данных с производительностью диска напрямую связан параметр *среднего времени доступа*. Он определяет интервал времени, необходимый для поиска нужных данных, и зависит от скорости вращения диска. Для дисков, вращающихся с частотой 5400 об/мин, среднее время доступа составляет 9–10 мкс, для дисков с частотой 7200 об/мин — 7–8 мкс. Изделия более высокого уровня обеспечивают среднее время доступа к данным 4–6 мкс.

Дисковод гибких дисков

Информация на жестком диске может храниться годами, однако иногда требуется ее перенос с одного компьютера на другой. Несмотря на свое название, жесткий диск является весьма хрупким прибором, чувствительным к перегрузкам, ударам и толчкам. Теоретически, переносить информацию с одного рабочего места на другое путем переноса жесткого диска возможно, и в некоторых случаях так и поступают, но все-таки этот прием считается нетехнологичным, поскольку требует особой аккуратности и определенной квалификации.

Для оперативного переноса небольших объемов информации используют так называемые *гибкие магнитные диски* (дискеты), которые вставляют в специальный накопитель — *дисковод*. Приемное отверстие накопителя находится на лицевой панели системного блока. Правильное направление подачи гибкого диска отмечено стрелкой на его пластиковом кожухе.

Основными параметрами гибких дисков являются: технологический размер (измеряется в дюймах), плотность записи (измеряется в кратных единицах) и полная емкость.

Первый компьютер IBM PC (родоначальник платформы) был выпущен в 1981 году. К нему можно было подключить внешний накопитель, использующий односторонние гибкие диски диаметром 5,25 дюйма. Емкость диска составляла 160 Кбайт. В следующем году появились аналогичные двусторонние диски емкостью 320 Кбайт. Начиная с 1984 года выпускались гибкие диски 5,25 дюйма высокой плотности (1,2 Мбайт). В наши дни диски размером 5,25 дюйма не

используются, так что производство и применение соответствующих дисководов практически прекратилось с середины 90-х годов.

Гибкие диски размером 3,5 дюйма выпускают с 1980 года. Односторонний диск *обычной плотности* имел емкость 180 Кбайт, двусторонний — 360 Кбайт, а *двусторонний двойной плотности* — 720 Кбайт. Ныне стандартными считают диски *высокой плотности* размером 3,5 дюйма. Они имеют емкость 1440 Кбайт (1,4 Мбайт) и маркируются буквами *HD* (high density — высокая плотность).

С нижней стороны гибкий диск имеет центральную втулку, которая захватывается шпинделем дисковода и приводится во вращение. Магнитная поверхность прикрыта сдвигающейся шторкой для защиты от влаги, грязи и пыли. Если на гибком диске записаны ценные данные, его можно защитить от стирания и перезаписи, сдвинув защитную задвижку так, чтобы образовалось открытое отверстие. Для разрешения записи задвижку перемещают в обратную сторону и перекрывают отверстие. В некоторых случаях для безусловной защиты информации на диске задвижку выламывают физически, но и в этом случае разрешить запись на диск можно, если, например, заклеить образовавшееся отверстие тонкой полоской липкой ленты.

Гибкие диски считаются малонадежными носителями информации. Пыль, грязь, влага, температурные перепады и внешние электромагнитные поля очень часто становятся причиной частичной или полной утраты данных, хранившихся на гибком диске. Поэтому использовать гибкие диски в качестве основного средства хранения информации недопустимо. Их используют только для транспортировки информации или в качестве дополнительного (резервного) средства хранения.

При передаче данных на гибком носителе следует придерживаться следующих правил этикета.

1. Все данные передаются в двух экземплярах.
2. Данные не удаляются с жесткого диска до тех пор, пока потребитель не подтвердил их благополучное получение, например по телефону.

При использовании гибких носителей в качестве резервного средства хранения данных следует придерживаться следующих рекомендаций.

1. Если эти данные неизменяемые, следует создать одну копию на гибком носителе, но не удалять данные с жесткого диска. Если данные с жесткого диска следует удалить, количество копий, закладываемых на хранение, должно быть не менее двух.
2. Если резервируемые данные подлежат периодическому изменению, то с жесткого диска их не удаляют, а количество резервных копий на гибких дисках должно быть не менее двух. Для этих копий устраивают периодическую ротацию с заданной периодичностью. Например, в конце первой рабочей недели копируют данные с жесткого диска на первый резервный комплект, а в конце второй недели — на второй резервный комплект, после чего еженедельно производят ротацию резервных комплектов.

При получении данных на гибком диске следует придерживаться следующих рекомендаций.

1. До начала работы с данными диск следует проверить антивирусными программными средствами. Среди вредоносных программ есть такие, которые поражают не только файлы программ и данных, но и носители информации. Даже «чистый» гибкий диск может содержать так называемые «загрузочные вирусы».
2. С данными, поставленными на гибком диске, работать не рекомендуется. Это не только непроизводительно, но и небезопасно (для данных). Прежде всего следует скопировать полученные данные на жесткий диск компьютера, после чего работать только с жестким диском.
3. Даже если работа с полученными данными в ближайшее время не предполагается, все равно их следует скопировать на жесткий диск немедленно после получения, так как во время хранения гибкого диска данные могут быть утрачены.
4. Правила делового этикета требуют немедленно после копирования данных с гибкого диска на жесткий оповестить лицо, предоставившее гибкий диск, о том, что прием данных состоялся. Это позволит ему сознательно распорядиться своими резервными копиями.

В новых компьютерах происходит постепенный отказ и от этого типа носителей, которые вытесняются записывающими дисковыми CD-RW.

Дисководы оптических дисков

В период 1994–1995 годов в базовую конфигурацию персональных компьютеров перестали включать дисководы гибких дисков диаметром 5,25 дюйма, но вместо них стандартной стала считаться установка дисководов CD-ROM, имеющего такие же внешние размеры.

Аббревиатура *CD-ROM* (Compact Disc Read-Only Memory) переводится на русский язык как *постоянное запоминающее устройство на основе компакт-диска*. Принцип действия этого устройства состоит в считывании числовых данных с помощью лазерного луча, отражающегося от поверхности диска (рис. 3.4). Цифровая запись на компакт-диске отличается от записи на магнитных дисках очень высокой плотностью, и стандартный компакт-диск может хранить примерно 650 Мбайт данных.

Практически все современные дисководы для работы с оптическими носителями поддерживают работу с так называемыми DVD-дисками (Digital Versatile Disc — цифровой многоцелевой диск). Стандартный DVD-диск содержит до 4,7 Гбайт информации. Существуют и более емкие оптические носители информации, например Blu-Ray Disc, емкость которых доходит до нескольких сотен гигабайт. Если компьютер оснащен более современным оптическим дисководом, обычно он поддерживает и носители предыдущих поколений. Более современные диски имеют те же размеры, что и CD, но большую плотность записи, нередко для хранения данных используют несколько слоев и две стороны диска.

Дисководы для чтения и записи DVD-дисков называют соответственно DVD-дисковыми, для работы с Blu-Ray Disc — BD-дисковыми.

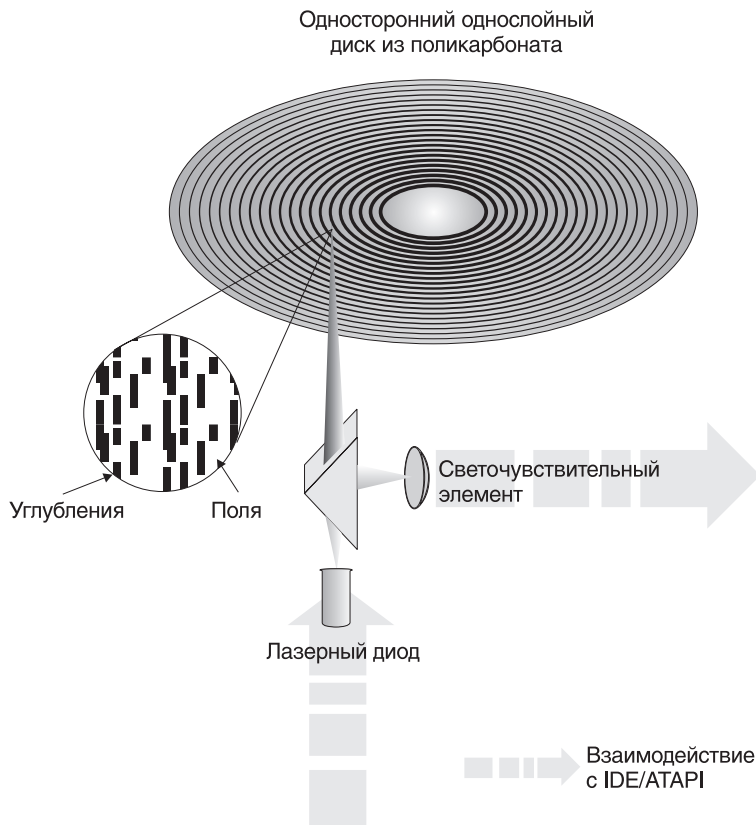


Рис. 3.4. Принцип действия дисководов CD-ROM

Большие объемы данных характерны для *мультимедийной информации* (графика, музыка, видео), поэтому CD/DVD/BD-дисководы относят к аппаратным средствам мультимедиа. Программные продукты, распространяемые на компакт-дисках, называют *мультимедийными изданиями*. Сегодня мультимедийные издания завоевывают все более прочное место среди других традиционных видов изданий. Так, например, существуют книги, альбомы, энциклопедии и даже периодические издания (электронные журналы), выпускаемые на *оптических дисках*.

Основным недостатком стандартных оптических дисководов является невозможность записи данных, но параллельно с ними сегодня существуют и устройства записи оптических дисков — дисководы CD/DVD/BD-RW. Для записи используются специальные заготовки. Некоторые из них допускают только однократную запись (после записи диск превращается в обычный оптический диск, доступный только для чтения), другие позволяют стереть ранее записанную информацию и выполнить запись заново.

Основным параметром оптических дисководов является скорость чтения данных. Она измеряется в кратных долях. За единицу измерения принята скорость чтения музыкальных компакт-дисков, составляющая в пересчете на данные

150 Кбайт/с. Таким образом, CD-ROM-дисковод с удвоенной скоростью чтения обеспечивает производительность 300 Кбайт/с, с учетверенной скоростью — 600 Кбайт/с и т. д. В настоящее время наибольшее распространение имеют устройства чтения CD-ROM с производительностью $48\times$ – $56\times$. Для заготовок, рассчитанных на однократную запись, скорость записи в соответствующих устройствах не уступает скорости чтения. Для заготовок многократной записи скорость записи может составлять $12\times$ – $24\times$.

Дисководы для работы с DVD- и BD-дисками обеспечивают более высокие скорости передачи данных при тех же скоростях вращения диска за счет более высокой плотности носителей.

Так, единица измерения скорости для DVD-диска составляет около 1352 Кбайт/с, то есть 1,32 Мбайт/с. Для BD-дисков тот же показатель равен примерно 4,5 Мбайт/с.

Видеокарта (видеоадаптер)

Совместно с монитором *видеокарта* образует *видеоподсистему* персонального компьютера. Видеокарта не всегда была компонентом ПК. На заре развития персональной вычислительной техники в общей области оперативной памяти существовала небольшая выделенная *экранная область памяти*, в которую процессор заносил данные об изображении. Специальный *контроллер экрана* считывал данные о яркости отдельных точек экрана из ячеек памяти этой области и в соответствии с ними управлял разверткой горизонтального луча электронной пушки монитора.

С переходом от черно-белых мониторов к цветным и с увеличением *разрешения экрана* (количества точек по вертикали и горизонтали) области видеопамати стало недостаточно для хранения графических данных, а процессор перестал справляться с построением и обновлением изображения. Тогда и произошло выделение всех операций, связанных с управлением экраном, в отдельный блок, получивший название *видеоадаптер*. Физически видеоадаптер выполнен в виде отдельной *дочерней платы*, которая вставляется в один из слотов материнской платы и называется *видеокартой*. Видеоадаптер взял на себя функции *видеоконтроллера*, *видеопроцессора* и *видеопамати*.

За время существования персональных компьютеров сменилось несколько стандартов видеоадаптеров: MDA (монохромный); CGA (4 цвета); EGA (16 цветов); VGA (256 цветов). В настоящее время применяются видеоадаптеры *SVGA*, обеспечивающие по выбору воспроизведение до 16,7 миллиона цветов с возможностью произвольного выбора разрешения экрана из стандартного ряда значений (640×480 , 800×600 , 1024×768 , 1152×864 ; 1280×1024 точек и далее).

Разрешение экрана является одним из важнейших параметров видеоподсистемы. Чем оно выше, тем больше информации можно отобразить на экране, но тем меньше размер каждой отдельной точки и соответственно тем меньше видимый размер элементов изображения. Использование завышенного разрешения на мониторе малого размера приводит к тому, что элементы изображения становятся неразборчивыми и работа с документами и программами вызывает утомление органов зрения. Использование заниженного разрешения приводит к тому, что элементы

изображения становятся крупными, но на экране их располагается очень мало. Если программа имеет сложную систему управления и большое число экранных элементов, они не полностью помещаются на экране. Это приводит к снижению производительности труда и неэффективной работе.

Таким образом, для каждого размера монитора существует свое оптимальное разрешение экрана, которое должен обеспечивать видеоадаптер (табл. 3.1). При качественном мониторе, хорошем зрении и ограниченном времени работы за компьютером разрешение можно увеличить на одну ступень.

Таблица 3.1. Разрешение экрана монитора

Размер монитора	Оптимальное разрешение экрана	Примечание
ЭЛТ 15 дюймов или ЖК менее 15 дюймов	800 × 600	Минимальное разрешение для современных программ
ЖК 15 дюймов или ЭЛТ 17 дюймов	1024 × 768	Типичное разрешение для современных программ
ЖК 17 дюймов или ЭЛТ 19 дюймов	1280 × 1024	

Большинство современных прикладных и развлекательных программ рассчитано на работу с разрешением экрана 800 × 600 и более. Именно поэтому сегодня минимально приемлемый размер монитора составляет 15 дюймов. Для работы с документами, подготовленными для печати на стандартных листах бумаги формата А4, необходимо экранное разрешение не менее 1024 × 768 и соответственно размер монитора в 17 дюймов.

Для работы в Интернете параметр разрешения зависит от способа оформления веб-страниц. Современные веб-страницы рассчитаны на работу с разрешением экрана 1024 × 768 и более.

Для большинства прикладных программ оптимальным также является разрешение 1024 × 768 и более, хотя в случае необходимости программы, как правило, допускают настройку своих панелей управления, делающую возможной работу в разрешении 800 × 600. Надо понимать, что при этом снижается производительность труда.

Таким образом, в настоящее время для работы с документами и службами Интернета наиболее приемлем ЖК-монитор размером более 17 дюймов. Размеры экранов более 20 дюймов и разрешения выше, чем 1024 × 768, применяют при работе с компьютерной графикой, системами автоматизированного проектирования и системами компьютерной верстки изданий.

Цветовое разрешение (глубина цвета) определяет количество различных оттенков, которые может принимать отдельная точка экрана. Максимально возможное цветовое разрешение зависит от свойств видеоадаптера и, в первую очередь, от количества установленной на нем видеопамяти. Кроме того, оно зависит и от установленного разрешения экрана. При высоком разрешении экрана на каждую точку изображения приходится отводить меньше места в видеопамяти, так что информация о цветах вынужденно оказывается более ограниченной.

В зависимости от заданного экранного разрешения и глубины цвета размер буфера кадра видеопамяти можно определить по следующей формуле:

$$P = \frac{(m - n)b}{8},$$

где P — необходимый объем памяти видеоадаптера; m — горизонтальное разрешение экрана (точек); n — вертикальное разрешение экрана (точек); b — разрядность кодирования цвета (бит).

Минимальное требование по глубине цвета на сегодняшний день — 256 цветов, хотя большинство программ требуют не менее 65 тыс. цветов (режим High Color). Наиболее комфортная работа достигается при глубине цвета 16,7 млн цветов (режим True Color).

Работа в полноцветном режиме True Color с высоким экранным разрешением требует значительных размеров видеопамяти. Современные видеоадаптеры способны также выполнять функции обработки изображения, снижая нагрузку на центральный процессор ценой дополнительных затрат видеопамяти. Объем видеопамяти, установленной на видеоадаптер, сегодня определяется не размером буфера кадра, а необходимостью выполнения подобных дополнительных операций и обычно составляет от нескольких сотен мегабайт до нескольких гигабайт.

Видеоускорение — одно из свойств видеоадаптера, которое заключается в том, что часть операций по построению изображений может происходить без выполнения математических вычислений в основном процессоре компьютера, а чисто аппаратным путем — преобразованием данных в микросхемах *видеоускорителя*. Видеоускорители обычно входят в состав видеоадаптера (в таких случаях говорят о том, что видеокарта обладает функциями аппаратного ускорения).

Различают два типа видеоускорителей — ускорители плоской (2D) и трехмерной (3D) графики. Первые наиболее эффективны для работы с прикладными программами, использующими стандартный интерфейс (обычно офисного применения), и оптимизированы для операционной системы Windows, а вторые ориентированы на работу мультимедийных развлекательных программ, в первую очередь компьютерных игр, и профессиональных программ обработки трехмерной графики. Обычно в этих случаях используют разные математические принципы автоматизации графических операций. Все современные видеокарты обладают функциями и двумерного, и трехмерного ускорения.

Звуковая карта

Звуковая карта явилась одним из наиболее поздних усовершенствований персонального компьютера. Она устанавливается в один из разъемов материнской платы в виде дочерней карты и выполняет вычислительные операции, связанные с обработкой звука, речи, музыки. Звук воспроизводится через внешние звуковые колонки, подключаемые к выходу звуковой карты. Специальный разъем позволяет отправить звуковой сигнал на внешний усилитель. Имеется также разъем для подключения микрофона, что позволяет записывать речь или музыку и сохранять их на жестком диске для последующей обработки и использования.

Основным параметром звуковой карты является *разрядность*, определяющая количество битов, используемых при преобразовании сигналов из аналоговой в цифровую форму, и наоборот. Чем выше разрядность, тем меньше погрешность, связанная с оцифровкой, тем выше качество звучания. Минимальным требованием сегодняшнего дня являются 16 разрядов, а наибольшее распространение имеют 32-разрядные и 64-разрядные устройства.

В области воспроизведения звука наиболее сложно обстоит дело со стандартизацией. В отсутствие единых централизованных стандартов, стандартом де-факто стали устройства, совместимые с устройством Sound Blaster, торговая марка на которое принадлежит компании Creative Labs.

В последнее время обработка звука рассматривается как относительно простая операция, которую, в связи с возросшей мощностью процессора, можно возложить и на него. В отсутствие повышенных требований к качеству звука можно использовать *интегрированные звуковые системы*, в которых функции обработки звука выполняются центральным процессором и микросхемами материнской платы. В этом случае колонки или иное устройство воспроизведения звука подключается к гнездам, установленным непосредственно на материнской плате.

3.3. Системы, расположенные на материнской плате

Оперативная память

Оперативная память (RAM — Random Access Memory) — это массив кристаллических ячеек, способных хранить данные. Существует много различных типов оперативной памяти, но с точки зрения физического принципа действия различают *динамическую память* (DRAM) и *статическую память* (SRAM).

Ячейки динамической памяти (DRAM) можно представить в виде микроконденсаторов, способных накапливать заряд на своих обкладках. Это наиболее распространенный и экономически доступный тип памяти. Недостатки этого типа связаны, во-первых, с тем, что как при заряде, так и при разряде конденсаторов неизбежны переходные процессы, то есть запись данных происходит сравнительно медленно. Второй важный недостаток связан с тем, что заряды ячеек имеют свойство рассеиваться в пространстве, причем весьма быстро. Если оперативную память постоянно не «подзаряжать», утрата данных происходит через несколько сотых долей секунды. Для борьбы с этим явлением в компьютере происходит постоянная *регенерация* (*освежение, подзарядка*) ячеек оперативной памяти. Регенерация осуществляется несколько десятков раз в секунду и вызывает непроизводительный расход ресурсов вычислительной системы.

Ячейки статической памяти (SRAM) можно представить как электронные микроэлементы — *триггеры*, состоящие из нескольких транзисторов. В триггере хранится не заряд, а состояние (*включен/выключен*), поэтому этот тип памяти обеспечивает более высокое быстродействие, хотя технологически он сложнее и, соответственно, дороже.

Микросхемы динамической памяти используют в качестве основной оперативной памяти компьютера. Микросхемы статической памяти используют в качестве

вспомогательной памяти (так называемой *кэш-памяти*), предназначенной для оптимизации работы процессора.

Каждая ячейка памяти имеет свой адрес, который выражается числом. В большинстве современных процессоров предельный размер адреса обычно составляет 32 разряда, а это означает, что всего независимых адресов может быть 2^{32} . Одна адресуемая ячейка содержит восемь двоичных ячеек, в которых можно сохранить 8 бит, то есть один байт данных.

Таким образом, в современных компьютерах возможна *непосредственная адресация* к полю памяти размером 2^{32} байт = 4 Гбайт. Однако это отнюдь не означает, что именно столько оперативной памяти непременно должно быть в компьютере. Предельный размер поля оперативной памяти, установленной в компьютере, определяется микропроцессорным комплектом (*чипсетом*) материнской платы и обычно не может превосходить нескольких гигабайт. Минимальный объем памяти определяется требованиями операционной системы и для современных компьютеров составляет 1 Гбайт.

Представление о том, сколько оперативной памяти *должно быть* в типовом компьютере, непрерывно меняется. В середине 80-х годов поле памяти размером 1 Мбайт казалось огромным, в начале 90-х годов достаточным считался объем 4 Мбайт, к середине 90-х годов он увеличился до 8 Мбайт, а затем и до 16 Мбайт. Сегодня типичным считается размер оперативной памяти в 1 Гбайт, но тенденция к росту сохраняется.

Оперативная память в компьютере размещается на стандартных панельках, называемых *модулями*. Модули оперативной памяти вставляют в соответствующие разъемы на материнской плате. Если к разъемам есть удобный доступ, то операцию можно выполнять своими руками. Если удобного доступа нет, может потребоваться неполная разборка узлов системного блока, и в таких случаях операцию поручают специалистам.

В современных компьютерах обычно применяют три типа модулей оперативной памяти. Модули памяти SDRAM (DIMM-модули) сегодня уже считаются устаревшими и используются в компьютерах прошлых поколений. Наиболее распространены модули типа DDR SDRAM (DDR DIMM), а также DDR2 и DDR3, обеспечивающие более быстрый доступ к памяти.

Основными характеристиками модулей оперативной памяти являются объем памяти и скорость передачи данных. Сегодня наиболее распространены модули объемом 512 Мбайт — 2 Гбайт. Скорость передачи данных определяет максимальную пропускную способность памяти (в мегабайтах в секунду или гигабайтах в секунду) в оптимальном режиме доступа. При этом учитывается время доступа к памяти, ширина шины и дополнительные возможности, такие как передача нескольких сигналов за один такт работы. Одинаковые по объему модули могут иметь разные скоростные характеристики.

Иногда в качестве определяющей характеристики памяти используют *время доступа*. Оно измеряется в миллиардных долях секунды (наносекундах, нс). Для современных модулей памяти это значение может составлять 5 нс, а для особо быстрой памяти, используемой в основном в видеокартах, — снижаться до 2–3 нс.

Процессор

Процессор — основная микросхема компьютера, в которой и производятся все вычисления. Конструктивно процессор состоит из ячеек, похожих на ячейки оперативной памяти, но в этих ячейках данные могут не только храниться, но и изменяться. Внутренние ячейки процессора называют *регистрами*. Важно также отметить, что данные, попавшие в некоторые регистры, рассматриваются не как данные, а как команды, управляющие обработкой данных в других регистрах. Среди регистров процессора есть и такие, которые в зависимости от своего содержания способны модифицировать исполнение команд. Таким образом, управляя засылкой данных в разные регистры процессора, можно управлять обработкой данных. На этом и основано исполнение программ.

С остальными устройствами компьютера, и в первую очередь с оперативной памятью, процессор связан несколькими группами проводников, называемых *шинами*. Основных шин три: *шина данных*, *адресная шина* и *командная шина*.

Адресная шина. У процессоров семейства Pentium адресная шина 32-разрядная, то есть состоит из 32 параллельных проводников. В зависимости от того, есть напряжение на какой-то из линий или нет, говорят, что на этой линии выставлена единица или ноль. Комбинация из 32 нулей и единиц образует 32-разрядный адрес, указывающий на одну из ячеек оперативной памяти. К ней и подключается процессор для копирования данных из ячейки в один из своих регистров.

Шина данных. По этой шине происходит копирование данных из оперативной памяти в регистры процессора и обратно. В современных персональных компьютерах шина данных, как правило, 64-разрядная, то есть состоит из 64 линий, по которым за один раз на обработку поступают сразу 8 байтов.

Шина команд. Для того чтобы процессор мог обрабатывать данные, ему нужны команды. Он должен знать, что следует сделать с теми байтами, которые хранятся в его регистрах. Эти команды поступают в процессор тоже из оперативной памяти, но не из тех областей, где хранятся массивы данных, а оттуда, где хранятся программы. Команды тоже представлены в виде байтов. Самые простые команды укладываются в один байт, однако есть и такие, для которых нужно два, три и более байтов. В большинстве современных процессоров шина команд 32-разрядная, хотя существуют 64-разрядные процессоры и даже 128-разрядные.

Система команд процессора. В процессе работы процессор обслуживает данные, находящиеся в его регистрах, в поле оперативной памяти, а также данные, находящиеся во внешних портах процессора. Часть данных он интерпретирует непосредственно как данные, часть данных — как адресные данные, а часть — как команды. Совокупность всех возможных команд, которые может выполнить процессор над данными, образует так называемую *систему команд процессора*. Процессоры, относящиеся к одному семейству, имеют одинаковые или близкие системы команд. Процессоры, относящиеся к разным семействам, различаются по системе команд и невзаимозаменяемы.

Процессоры с расширенной и сокращенной системой команд. Чем шире набор системных команд процессора, тем сложнее его архитектура, тем длиннее формальная запись команды (в байтах), тем выше средняя продолжительность

исполнения одной команды, измеренная в тактах работы процессора. Так, например, система команд процессоров семейства Pentium в настоящее время насчитывает более тысячи различных команд. Такие процессоры называют *процессорами с расширенной системой команд* — CISC-процессорами (CISC — Complex Instruction Set Computing).

В противоположность CISC-процессорам в середине 80-х годов появились процессоры архитектуры RISC с сокращенной системой команд (RISC — Reduced Instruction Set Computing). При такой архитектуре количество команд в системе намного меньше и каждая из них выполняется намного быстрее. Таким образом, программы, состоящие из простейших команд, выполняются этими процессорами много быстрее. Обратная сторона сокращенного набора команд состоит в том, что сложные операции приходится эмулировать далеко не эффективной последовательностью простейших команд сокращенного набора.

В результате конкуренции между двумя подходами к архитектуре процессоров сложилось следующее распределение их сфер применения:

- ❑ CISC-процессоры используют в универсальных вычислительных системах;
- ❑ RISC-процессоры используют в специализированных вычислительных системах или устройствах, ориентированных на выполнение единообразных операций.

Персональные компьютеры платформы IBM PC ориентированы на использование CISC-процессоров.

Совместимость процессоров. Если два процессора имеют одинаковую систему команд, то они полностью совместимы на программном уровне. Это означает, что программа, написанная для одного процессора, может исполняться и другим процессором. Процессоры, имеющие разные системы команд, как правило, несовместимы или ограниченно совместимы на программном уровне.

Группы процессоров, имеющих ограниченную совместимость, рассматривают как *семейства процессоров*. Так, например, все процессоры Intel Pentium относятся к так называемому семейству x86. Родоначальником этого семейства был 16-рядный процессор Intel 8086, на базе которого собиралась первая модель компьютера IBM PC. Впоследствии выпускались процессоры Intel 80286, Intel 80386, Intel 80486, несколько моделей Intel Pentium; несколько моделей Intel Pentium MMX, модели Intel Pentium Pro, Intel Pentium II, Intel Celeron, Intel Xeon, Intel Pentium III, Intel Pentium 4, Intel Core и другие. Все эти модели, и не только они, а также многие модели процессоров компании AMD и некоторых других производителей относятся к семейству x86 (или к семейству x86-64) и обладают совместимостью по принципу «сверху вниз».

Принцип совместимости «сверху вниз» — это пример неполной совместимости, когда каждый новый процессор «понимает» все команды своих предшественников, но не наоборот. Это естественно, поскольку двадцать лет назад разработчики процессоров не могли предусмотреть систему команд, нужную для современных программ. Благодаря такой совместимости на современном компьютере можно выполнять любые программы, созданные в последние десятилетия для любого из предшествующих компьютеров, принадлежащего той же аппаратной платформе.

Основные параметры процессоров. Основными параметрами процессоров являются: *рабочее напряжение, разрядность, рабочая тактовая частота, коэффициент внутреннего умножения тактовой частоты* и размер *кэш-памяти*.

Рабочее напряжение процессора обеспечивает материнская плата, поэтому разным маркам процессоров соответствуют разные материнские платы (их надо выбирать совместно). По мере развития процессорной техники происходит постепенное понижение рабочего напряжения. Ранние модели процессоров x86 имели рабочее напряжение 5 В. С переходом к процессорам Intel Pentium оно было понижено до 3,3 В, а в настоящее время оно составляет менее 2 В. Понижение рабочего напряжения позволяет уменьшить расстояния между структурными элементами в кристалле процессора до десятитысячных долей миллиметра, не опасаясь электрического пробоя. Пропорционально квадрату напряжения уменьшается и тепловыделение в процессоре, а это позволяет увеличивать его производительность без угрозы перегрева.

Разрядность процессора показывает, сколько бит данных он может принять и обработать в своих регистрах за один раз (за один такт). Первые процессоры x86 были 16-разрядными. Начиная с процессора 80386, они имеют 32-разрядную архитектуру. Подавляющее большинство современных процессоров являются 64-разрядными, но они полностью поддерживают архитектуру x86.

В основе работы процессора лежит тот же тактовый принцип, что и в обычных часах. Исполнение каждой команды занимает определенное количество тактов. В настенных часах такты колебаний задает маятник; в ручных механических часах их задает пружинный маятник; в электронных часах для этого есть колебательный контур, задающий такты строго определенной частоты. В персональном компьютере тактовые импульсы задает одна из микросхем, входящая в микропроцессорный комплект (чипсет), расположенный на материнской плате. Чем выше частота тактов, поступающих на процессор, тем больше команд он может исполнить в единицу времени, тем выше его производительность. Первые процессоры x86 могли работать с частотой не выше 4,77 МГц, а сегодня *рабочие частоты* некоторых процессоров уже превосходят 3 миллиарда тактов в секунду (3 ГГц).

Тактовые сигналы процессор получает от материнской платы, которая, в отличие от процессора, представляет собой не кристалл кремния, а большой набор проводников и микросхем. По чисто физическим причинам материнская плата не может работать со столь высокими частотами, как процессор. Сегодня базовая частота материнской платы составляет 100–200 МГц. Для получения более высоких частот в процессоре происходит *внутреннее умножение частоты*. Коэффициент внутреннего умножения в современных процессорах может достигать 10–20 и выше.

Обмен данными внутри процессора происходит в несколько раз быстрее, чем обмен с другими устройствами, например с оперативной памятью. Для того чтобы уменьшить количество обращений к оперативной памяти, внутри процессора создают буферную область — так называемую *кэш-память*. Это как бы «сверх-оперативная память». Когда процессору нужны данные, он сначала обращается в кэш-память, и только если там нужных данных нет, происходит его обращение в оперативную память. Принимая блок данных из оперативной памяти, процессор

заносит его одновременно и в кэш-память. «Удачные» обращения в кэш-память называют *попаданиями в кэш*. Процент попаданий тем выше, чем больше размер кэш-памяти, поэтому высокопроизводительные процессоры комплектуют повышенным объемом кэш-памяти.

Нередко кэш-память распределяют по нескольким уровням. Кэш первого уровня выполняется в том же кристалле, что и сам процессор, и имеет объем порядка десятков килобайт. Кэш второго уровня находится либо в кристалле процессора, либо в том же узле, что и процессор, хотя и выполняется на отдельном кристалле. Кэш-память первого и второго уровня работает на частоте, согласованной с частотой ядра процессора.

Кэш-память третьего уровня выполняют на быстродействующих микросхемах типа SRAM и размещают на материнской плате вблизи процессора. Ее объемы могут достигать нескольких мегабайт, но работает она на частоте материнской платы.

Микросхема ПЗУ и система BIOS

В момент включения компьютера в его оперативной памяти нет ничего — ни данных, ни программ, поскольку оперативная память не может ничего хранить без подзарядки ячеек более сотых долей секунды, но процессору нужны команды, в том числе и в первый момент после включения. Поэтому сразу после включения на адресной шине процессора выставляется стартовый адрес. Это происходит аппаратно, без участия программ (всегда одинаково). Процессор обращается по выставленному адресу за своей первой командой и далее начинает работать по программам.

Этот исходный адрес не может указывать на оперативную память, в которой пока ничего нет. Он указывает на другой тип памяти — *постоянное запоминающее устройство* (ПЗУ). Микросхема ПЗУ способна длительное время хранить информацию, даже когда компьютер выключен. Программы, находящиеся в ПЗУ, называют «защитыми» — их записывают туда на этапе изготовления микросхемы.

Комплект программ, находящихся в ПЗУ, образует *базовую систему ввода-вывода* (BIOS — Basic Input Output System). Основное назначение программ этого пакета состоит в том, чтобы проверить состав и работоспособность компьютерной системы и обеспечить взаимодействие с клавиатурой, монитором, жестким диском и дисководом гибких дисков. Программы, входящие в BIOS, позволяют нам наблюдать на экране диагностические сообщения, сопровождающие запуск компьютера, а также вмешиваться в ход запуска с помощью клавиатуры.

Энергонезависимая память CMOS

Выше мы отметили, что работа таких стандартных устройств, как клавиатура, может обслуживаться программами, входящими в BIOS, но такими средствами нельзя обеспечить работу со всеми возможными устройствами. Так, например, изготовители BIOS абсолютно ничего не знают о параметрах наших жестких и гибких дисков, им не известны ни состав, ни свойства произвольной вычислительной системы. Для того чтобы начать работу с другим оборудованием, программы,

входящие в состав BIOS, должны знать, где можно найти нужные параметры. По очевидным причинам их нельзя хранить ни в оперативной памяти, ни в постоянном запоминающем устройстве.

Специально для этого на материнской плате есть микросхема «энергонезависимой памяти», по технологии изготовления называемая CMOS. От оперативной памяти она отличается тем, что ее содержимое не стирается во время выключения компьютера, а от ПЗУ она отличается тем, что данные в нее можно заносить и изменять самостоятельно, в соответствии с тем, какое оборудование входит в состав системы. Эта микросхема постоянно подпитывается от небольшой аккумуляторной батарейки, расположенной на материнской плате. Заряда этой батарейки хватает на то, чтобы микросхема не теряла данные, даже если компьютер не будет включать месяцами.

В микросхеме CMOS хранятся данные о гибких и жестких дисках, о процессоре, о некоторых других устройствах материнской платы. Тот факт, что компьютер четко отслеживает время и календарь (даже и в выключенном состоянии), тоже связан с тем, что показания системных часов постоянно хранятся (и изменяются) в CMOS.

Таким образом, программы, записанные в BIOS, считывают данные о составе оборудования компьютера из микросхемы CMOS, после чего они могут выполнить обращение к жесткому диску, а в случае необходимости и к гибкому, и передать управление тем программам, которые там записаны.

Шинные интерфейсы материнской платы

Связь между всеми собственными и подключаемыми устройствами материнской платы выполняют ее шины и логические устройства, размещенные в микросхемах микропроцессорного комплекта (чипсета). От архитектуры этих элементов во многом зависит производительность компьютера.

ISA. Историческим достижением компьютеров платформы IBM PC стало внедрение почти двадцать лет назад архитектуры, получившей статус *промышленного стандарта ISA* (Industry Standard Architecture). Она не только позволила связать все устройства системного блока между собой, но и обеспечила простое подключение новых устройств через стандартные разъемы (слоты). Пропускная способность шины, выполненной по такой архитектуре, составляет до 5,5 Мбайт/с, но, несмотря на низкую пропускную способность, эта шина еще может использоваться в некоторых компьютерах для подключения сравнительно «медленных» внешних устройств, например звуковых карт и модемов.

EISA. Расширением стандарта ISA стал стандарт *EISA* (Extended ISA), отличающийся увеличенным разъемом и увеличенной производительностью (до 32 Мбайт/с). Как и ISA, в настоящее время данный стандарт считается устаревшим. После 2000 года выпуск материнских плат с разъемами ISA/EISA и устройств, подключаемых к ним, практически прекращен.

VLB. Название интерфейса переводится как *локальная шина стандарта VESA* (VESA Local Bus). Понятие «локальной шины» впервые появилось в конце 80-х годов. Оно связано тем, что при внедрении процессоров третьего и четвертого поколений (Intel 80386 и Intel 80486) частоты основной шины (в качестве основной

использовалась шина ISA/EISA) стало недостаточно для обмена между процессором и оперативной памятью. Локальная шина, имеющая повышенную частоту, связала между собой процессор и память в обход основной шины. Впоследствии в эту шину «врезали» интерфейс для подключения видеоадаптера, который тоже требует повышенной пропускной способности, — так появился стандарт *VLB*, который позволил поднять тактовую частоту локальной шины до 50 МГц и обеспечил пиковую пропускную способность до 130 Мбайт/с.

Основным недостатком интерфейса *VLB* стало то, что предельная частота локальной шины и, соответственно, ее пропускная способность зависят от числа устройств, подключенных к шине. Так, например, при частоте 50 МГц к шине может быть подключено только одно устройство (видеокарта). Для сравнения скажем, что при частоте 40 МГц возможно подключение двух, а при частоте 33 МГц — трех устройств. Активное использование шины *VLB* продолжалось очень недолго, она была вскоре вытеснена шиной *PCI*.

PCI. Интерфейс *PCI* (Peripheral Component Interconnect — стандарт подключения внешних компонентов) был введен в персональных компьютерах во времена процессора 80486 и первых версий Pentium. По своей сути это тоже интерфейс локальной шины, связывающей процессор с оперативной памятью, в которую врезаются разъемы для подключения внешних устройств. Для связи с основной шиной компьютера (ISA/EISA) используются специальные интерфейсные преобразователи — *мосты PCI* (*PCI Bridge*). В современных компьютерах функции моста *PCI* выполняют микросхемы микропроцессорного комплекта (чипсета).

Данный интерфейс поддерживает частоту шины 33 МГц и обеспечивает пропускную способность 132 Мбайт/с. Последние версии интерфейса поддерживают частоту до 66 МГц и обеспечивают производительность 264 Мбайт/с для 32-разрядных данных и 528 Мбайт/с для 64-разрядных данных.

Важным нововведением, реализованным этим стандартом, стала поддержка так называемого режима *plug-and-play*, впоследствии оформившегося в промышленный стандарт на *самоустанавливающиеся устройства*. Его суть состоит в том, что после физического подключения внешнего устройства к разъему шины *PCI* происходит обмен данными между устройством и материнской платой, в результате которого устройство автоматически получает номер используемого прерывания, адрес порта подключения и номер канала прямого доступа к памяти.

Конфликты между устройствами за обладание одними и теми же ресурсами (номерах прерываний, адресами портов и каналами прямого доступа к памяти) вызывают массу проблем у пользователей при установке устройств, подключаемых к шине *ISA*. С появлением интерфейса *PCI* и оформлением стандарта *plug-and-play* появилась возможность выполнять установку новых устройств с помощью автоматических программных средств — эти функции во многом были возложены на операционную систему.

FSB. Шина *PCI*, появившаяся в компьютерах на базе процессоров Intel Pentium как локальная шина, предназначенная для связи процессора с оперативной памятью, недолго оставалась в этом качестве. Сегодня она используется только как шина для подключения внешних устройств, а для связи процессора и памяти, начиная с процессора Intel Pentium Pro, используется специальная шина, получившая

название *Front Side Bus* (FSB). Эта шина работает на частоте 100–200 МГц. Частота шины FSB является одним из основных потребительских параметров — именно он и указывается в спецификации материнской платы. Современные типы памяти (DDR SDRAM, RDRAM) способны передавать несколько сигналов за один такт шины FSB, что повышает скорость обмена данными с оперативной памятью.

AGP. Видеоадаптер — устройство, требующее особенно высокой скорости передачи данных. Как при внедрении локальной шины VLB, так и при внедрении локальной шины PCI видеоадаптер всегда был первым устройством, «врезаемым» в новую шину. Когда параметры шины PCI перестали соответствовать требованиям видеоадаптеров, для них была разработана отдельная шина, получившая название *AGP* (Advanced Graphic Port — усовершенствованный графический порт). Частота этой шины соответствует частоте шины *PCI* (33 МГц или 66 МГц), но она имеет много более высокую пропускную способность за счет передачи нескольких сигналов за один такт. Число сигналов, передаваемых за один такт, указывается в виде множителя, например AGP 4× (в этом режиме скорость передачи достигает 1066 Мбайт/с). Последняя версия шины AGP имеет кратность 8×.

PCI Express. Эта шина пришла на смену шинам PCI и AGP и используется в современных компьютерах. Отличительной особенностью PCI Express является применение последовательного протокола передачи данных, который позволил достигнуть весьма высоких показателей производительности. Так, пропускная способность шины достигает 64 Гбайт/с. Существуют различные варианты PCI Express, которые различаются количеством соединений. Так, низкоскоростные устройства (такие, как модемы) могут использовать PCI Express X1 (пропускная способность «одинарной» линии составляет 0,5 Гбайт/с). Современные видеокарты обычно используют PCI Express X16.

PCMCIA (Personal Computer Memory Card International Association — стандарт международной ассоциации производителей плат памяти для персональных компьютеров). Этот стандарт определяет интерфейс подключения плоских карт памяти небольших размеров и используется в портативных персональных компьютерах.

USB (Universal Serial Bus — универсальная последовательная магистраль). Это одно из последних нововведений в архитектурах материнских плат. Этот стандарт определяет способ взаимодействия компьютера с периферийным оборудованием. Он позволяет подключать до 256 различных устройств, имеющих последовательный интерфейс. Устройства могут включаться цепочками (каждое следующее устройство подключается к предыдущему). Изначально производительность шины USB была относительно невелика, но ее вполне хватало для таких устройств, как клавиатура, мышь, модем, джойстик, принтер и т. п. Современные варианты USB поддерживают высокие скорости передачи данных, что делает ее универсальной шиной для подключения к ПК любых внешних устройств.

Удобство шины состоит в том, что она практически исключает конфликты между различным оборудованием, позволяет подключать и отключать устройства в «горячем режиме» (не выключая компьютер) и позволяет объединять несколько компьютеров в простейшую локальную сеть без применения специального оборудования и программного обеспечения.

Функции микропроцессорного комплекта (чипсета)

Параметры микропроцессорного комплекта (чипсета) в наибольшей степени определяют свойства и функции материнской платы. В настоящее время большинство чипсетов материнских плат выпускаются на базе двух микросхем, исторически получивших название «северный мост» и «южный мост».

«Северный мост» обычно управляет взаимосвязью процессора, оперативной памяти и порта AGP.

«Южный мост» называют также *функциональным контроллером*. Он выполняет функции контроллера жестких и гибких дисков, функции контроллера шины PCI, моста ISA — PCI, контроллера клавиатуры, мыши, шины USB и т. п.

У предыдущих поколений материнских плат связь между северным и южным мостом обеспечивала шина PCI, контроллер которой располагался в северном мосте. У современных материнских плат мосты соединены новой шиной повышенной производительности, а контроллер шины PCI находится в южном мосте вместе с контроллерами всех прочих устройств.

3.4. Периферийные устройства персонального компьютера

Периферийные устройства персонального компьютера подключаются к его интерфейсам и предназначены для выполнения вспомогательных операций. Благодаря им компьютерная система приобретает гибкость и универсальность.

По назначению периферийные устройства можно подразделить на:

- ☐ устройства ввода данных;
- ☐ устройства вывода данных;
- ☐ устройства хранения данных;
- ☐ устройства обмена данными.

Устройства ввода знаковых данных

Специальные клавиатуры. Клавиатура является основным устройством ввода данных. Специальные клавиатуры предназначены для повышения эффективности процесса ввода данных. Это достигается путем изменения формы клавиатуры, раскладки ее клавиш или метода подключения к системному блоку.

Клавиатуры, имеющие специальную форму, рассчитанную с учетом требований эргономики, называют *эргономичными клавиатурами*. Их целесообразно применять на рабочих местах, предназначенных для ввода большого количества знаковой информации. Эргономичные клавиатуры не только повышают производительность наборщика и снижают общее утомление в течение рабочего дня, но и снижают вероятность и степень развития ряда заболеваний, например туннельного синдрома кистей рук и остеохондроза верхних отделов позвоночника.

Раскладка клавиш стандартных клавиатур далека от оптимальной. Она сохранилась со времен ранних образцов механических пишущих машин. В настоящее время существует техническая возможность изготовления клавиатур с оптимизи-

рованной раскладкой и существуют образцы таких устройств (в частности, к ним относится *клавиатура Дворака*). Однако практическое внедрение клавиатур с нестандартной раскладкой находится под вопросом в связи с тем, что работе с ними надо учиться специально. На практике подобными клавиатурами оснащают только специализированные рабочие места.

По методу подключения к системному блоку различают *проводные* и *беспроводные* клавиатуры. Передача информации в беспроводных системах осуществляется инфракрасным лучом. Обычный радиус действия таких клавиатур составляет несколько метров. Источником сигнала является клавиатура.

Устройства командного управления

Специальные манипуляторы. Кроме обычной мыши существуют и другие типы манипуляторов, например: трекболы, пенмаусы, инфракрасные мыши.

Трекбол в отличие от мыши устанавливается стационарно, и его шарик приводится в движение ладонью руки. Преимущество трекбола состоит в том, что он не нуждается в гладкой рабочей поверхности, поэтому трекболы нашли широкое применение в портативных персональных компьютерах.

В последнее время, однако, в портативных компьютерах вместо трекболов используются *тачпады* — сенсорные пластины, реагирующие на движение пальца пользователя по поверхности. Удар пальцем по поверхности тачпада воспринимается как нажатие кнопки. Недостатком тачпадов является невысокая точность.

Пенмаус представляет собой аналог шариковой авторучки, на конце которой вместо пишущего узла установлен узел, регистрирующий величину перемещения.

Инфракрасная мышь отличается от обычной наличием устройства беспроводной связи с системным блоком.

Для компьютерных игр и в некоторых специализированных имитаторах применяют также манипуляторы рычажно-нажимного типа (*джойстики*) и аналогичные им *джойпады*, *геймпады* и *штурвально-педальные* устройства. Устройства этого типа подключаются к специальному порту, имеющемуся на звуковой карте, или к порту USB.

Устройства ввода графических данных

Для ввода графической информации используют *сканеры*, *графические планшеты* (*дигитайзеры*) и *цифровые фотокамеры*. Интересно отметить, что с помощью сканеров можно вводить и знаковую информацию. В этом случае исходный материал вводится в графическом виде, после чего обрабатывается специальными программными средствами (*программами распознавания образов*).

Планшетные сканеры. Планшетные сканеры предназначены для ввода графической информации с прозрачного или непрозрачного листового материала. Принцип действия этих устройств состоит в том, что луч света, отраженный от поверхности материала (или прошедший сквозь прозрачный материал), фиксируется специальными элементами, называемыми *приборами с зарядовой связью* (ПЗС). Обычно элементы ПЗС конструктивно оформляют в виде линейки, располагаемой по

ширине исходного материала. Перемещение линейки относительно листа бумаги выполняется механическим протягиванием линейки при неподвижной установке листа или протягиванием листа при неподвижной установке линейки.

Основными потребительскими параметрами планшетных сканеров являются:

- ☐ разрешающая способность;
- ☐ производительность;
- ☐ динамический диапазон;
- ☐ максимальный размер сканируемого материала.

Разрешающая способность планшетного сканера зависит от плотности размещения приборов ПЗС на линейке, а также от точности механического позиционирования линейки при сканировании. Типичный показатель для офисного применения: 600–1200 dpi (dpi — dots per inch, количество точек на дюйм). Для профессионального применения характерны показатели 1200–3000 dpi.

Производительность сканера определяется продолжительностью сканирования листа бумаги стандартного формата и зависит как от совершенства механической части устройства, так и от типа интерфейса, использованного для сопряжения с компьютером.

Динамический диапазон определяется логарифмом отношения яркости наиболее светлых участков изображения к яркости наиболее темных участков. Типовой показатель для сканеров офисного применения составляет 1,8–2,0, а для сканеров профессионального применения — от 2,5 (для непрозрачных материалов) до 3,5 (для прозрачных материалов).

Ручные сканеры. Принцип действия ручных сканеров в основном соответствует планшетным. Разница заключается в том, что протягивание линейки ПЗС в данном случае выполняется вручную. Равномерность и точность сканирования при этом обеспечиваются неудовлетворительно, и разрешающая способность ручного сканера составляет 150–300 dpi.

Барабанные сканеры. В сканерах этого типа исходный материал закрепляется на цилиндрической поверхности барабана, вращающегося с высокой скоростью. Устройства этого типа обеспечивают наивысшее разрешение (2400–5000 dpi) благодаря применению не ПЗС, а фотоэлектронных умножителей. Их используют для сканирования исходных изображений, имеющих высокое качество, но недостаточные линейные размеры (фотонегативов, слайдов и т. п.).

Сканеры форм. Предназначены для ввода данных со стандартных форм, заполненных механически или «от руки». Необходимость в этом возникает при проведении переписей населения, обработке результатов выборов и анализе анкетных данных. От сканеров форм не требуется высокой точности сканирования, но быстродействие играет повышенную роль и является основным потребительским параметром.

Штрих-сканеры. Эта разновидность ручных сканеров предназначена для ввода данных, закодированных в виде штрих-кода. Такие устройства имеют применение в розничной торговой сети.

Графические планшеты (дигитайзеры) предназначены для ввода художественной графической информации. Существует несколько различных принципов дей-

ствия графических планшетов, но в основе всех их лежит фиксация перемещения специального пера относительно планшета. Устройства удобны для художников и иллюстраторов, поскольку позволяют им создавать экранные изображения привычными приемами, наработанными для традиционных инструментов (карандаш, перо, кисть).

Цифровые фотокамеры. Как и сканеры, эти устройства воспринимают графические данные с помощью приборов с зарядовой связью, объединенных в прямоугольную матрицу. Основным параметром цифровых фотоаппаратов является разрешающая способность, которая напрямую связана с количеством ячеек ПЗС в матрице. Наилучшие потребительские модели в настоящее время имеют 2–16,6 млн ячеек ПЗС и, соответственно, обеспечивают разрешение изображения до 4992×3328 точек и выше. У профессиональных моделей эти параметры еще выше.

Устройства вывода данных

В качестве устройств вывода данных, дополнительных к монитору, используют печатающие устройства (принтеры), позволяющие получать копии документов на бумаге или прозрачном носителе. По принципу действия различают матричные, лазерные, светодиодные и струйные принтеры.

Матричные принтеры. Это простейшие печатающие устройства. Данные выводятся на бумагу в виде оттиска, образуемого при ударе цилиндрических стержней («иглодок») через красящую ленту. Качество печати матричных принтеров напрямую зависит от количества иглодок в печатающей головке. Наибольшее распространение имеют *9-игольчатые* и *24-игольчатые* матричные принтеры. Последние позволяют получать оттиски документов, практически не уступающие по качеству документам, исполненным на пишущей машинке. В настоящее время матричные принтеры считаются устаревшими и практически не выпускаются.

Производительность работы матричных принтеров оценивают по количеству печатаемых знаков в секунду (*cps — characters per second*). Обычными режимами работы матричных принтеров являются: *draft* — режим черновой печати, *normal* — режим обычной печати и режим *NLQ (Near Letter Quality)*, который обеспечивает качество печати, близкое к качеству пишущей машинки.

Лазерные принтеры обеспечивают высокое качество печати, не уступающее, а во многих случаях и превосходящее полиграфическое. Они отличаются также высокой скоростью печати, которая измеряется в страницах в минуту (*ppm — page per minute*). Как и в матричных принтерах, итоговое изображение формируется из отдельных точек.

Принцип действия лазерных принтеров следующий:

- ❑ в соответствии с поступающими данными лазерная головка испускает световые импульсы, которые отражаются от зеркала и попадают на поверхность светочувствительного барабана;
- ❑ горизонтальная развертка изображения выполняется вращением зеркала;
- ❑ участки поверхности светочувствительного барабана, получившие световой импульс, приобретают статический заряд;

- ❑ барабан при вращении проходит через контейнер, наполненный красящим составом (тонером), и тонер закрепляется на участках, имеющих статический заряд;
- ❑ при дальнейшем вращении барабана происходит контакт его поверхности с бумажным листом, в результате чего происходит перенос тонера на бумагу;
- ❑ лист бумаги с нанесенным на него тонером протягивается через нагревательный элемент, в результате чего частицы тонера спекаются и закрепляются на бумаге.

К основным параметрам лазерных принтеров относятся:

- ❑ разрешающая способность, dpi (dots per inch — точек на дюйм);
- ❑ производительность (страниц в минуту);
- ❑ формат используемой бумаги;
- ❑ объем собственной оперативной памяти.

При выборе лазерного принтера необходимо также учитывать параметр стоимости оттиска, то есть стоимость расходных материалов для получения одного печатного листа стандартного формата А4. К расходным материалам относятся тонер и барабан, который после печати определенного количества оттисков утрачивает свои свойства. В качестве единицы измерения используют *цент на страницу* (имеются в виду центы США). В настоящее время теоретический предел по этому показателю составляет порядка 1,0–1,5. На практике лазерные принтеры массового применения обеспечивают значения от 2,0 до 6,0.

Основное преимущество лазерных принтеров заключается в возможности получения высококачественных отпечатков. Уже модели среднего класса обеспечивают разрешение печати до 600 dpi, а профессиональные модели — до 1800 dpi и выше.

Светодиодные принтеры. Принцип действия светодиодных принтеров похож на принцип действия лазерных принтеров. Разница заключается в том, что источником света является не лазерная головка, а линейка светодиодов. Поскольку эта линейка расположена по всей ширине печатаемой страницы, отпадает необходимость в механизме формирования горизонтальной развертки и вся конструкция получается проще, надежнее и дешевле. Типичная величина разрешения печати для светодиодных принтеров составляет порядка 600 dpi.

Струйные принтеры. В струйных печатающих устройствах изображение на бумаге формируется из пятен, образующихся при попадании капель красителя на бумагу. Выброс микрокапель красителя происходит под давлением, которое развивается в печатающей головке за счет парообразования. В некоторых моделях капля выбрасывается щелчком в результате пьезоэлектрического эффекта — этот метод позволяет обеспечить более стабильную форму капли, близкую к сферической.

Качество печати изображения во многом зависит от формы капли и ее размера, а также от характера впитывания жидкого красителя поверхностью бумаги. В этих условиях особую роль играют вязкостные свойства красителя и свойства бумаги.

К положительным свойствам струйных печатающих устройств следует отнести относительно небольшое количество движущихся механических частей и, соответственно, простоту и надежность механической части устройства и его относительно низкую стоимость. Основным недостатком, по сравнению с лазерными принтерами,

является нестабильность получаемого разрешения, что ограничивает возможность их применения в черно-белой полутоновой печати.

Сегодня струйные принтеры нашли очень широкое применение в цветной печати. Благодаря простоте конструкции они намного превосходят цветные лазерные принтеры по показателю качество/цена. При разрешении выше 600 dpi они позволяют получать цветные оттиски, превосходящие по качеству цветные отпечатки, получаемые фотохимическими методами.

При выборе струйного принтера следует обязательно иметь в виду параметр стоимости печати одного оттиска. Хотя цена струйных печатающих устройств заметно ниже, чем лазерных, стоимость печати одного оттиска на них может быть в несколько раз выше.

Устройства хранения данных

Необходимость во внешних устройствах хранения данных возникает в двух случаях:

- когда на вычислительной системе обрабатывается больше данных, чем можно разместить на базовом жестком диске;
- когда данные имеют повышенную ценность и необходимо выполнять регулярное резервное копирование на внешнее устройство (копирование данных в пределах того же жесткого диска не является резервным и только создает иллюзию безопасности).

В настоящее время для внешнего хранения данных используют несколько типов устройств, использующих магнитные или магнитооптические носители.

Стримеры. Стримеры — это накопители на магнитной ленте. Их отличает сравнительно низкая цена. К недостаткам стримеров относят малую производительность (она связана, прежде всего, с тем, что магнитная лента — это устройство последовательного доступа) и недостаточную надежность (кроме электромагнитных наводок, ленты стримеров испытывают повышенные механические нагрузки и могут физически выходить из строя).

Емкость магнитных кассет (картриджей) для стримеров достигает нескольких десятков гигабайт. Дальнейшее повышение емкости за счет повышения плотности записи снижает надежность хранения, а повышение емкости за счет увеличения длины ленты сдерживается низким временем доступа к данным.

Накопители на съемных магнитных дисках. К этой категории относится несколько разных типов устройств, ни одно из которых так и не стало общепринятым стандартом. Например, ZIP-накопители выпускаются компанией Imomega, специализирующейся на создании внешних устройств для хранения данных. Устройство работает с дисковыми носителями, по размеру незначительно превышающими стандартные гибкие диски и имеющими емкость 100/250/750 Мбайт. Основным недостатком ZIP-накопителей является отсутствие их совместимости со стандартными гибкими дисками 3,5 дюйма. Такой совместимостью обладают устройства HiFD компании Sony. Они позволяют использовать как специальные носители емкостью 200 Мбайт, так и обычные гибкие диски. Распространение этих устройств сдерживается высокой ценой.

Накопители JAZ, как и ZIP-накопители, выпускаются компанией Iomega. По своим характеристикам JAZ-носитель приближается к жестким дискам, но в отличие от них является сменным. В зависимости от модели накопителя на одном диске можно разместить один или два гигабайта данных.

Магнитооптические устройства. Эти устройства получили широкое распространение в компьютерных системах высокого уровня благодаря своей универсальности. С их помощью решаются задачи резервного копирования, обмена данными и их накопления. Однако достаточно высокая стоимость приводов и носителей не позволяет отнести их к устройствам массового спроса.

В этом секторе параллельно развиваются 5,25- и 3,5-дюймовые накопители, носители для которых отличаются в основном форм-фактором и емкостью. Последнее поколение носителей формата 5,25" достигает емкости 5,2 Гбайт. Емкость носителей 3,5" несколько ниже, от 640 Мбайт до 2,3 Гбайт.

В перспективе ожидается появление накопителей заметно большего объема (до нескольких десятков гигабайт).

Флэш-диски. Это современное устройство хранения данных на основе энерго-независимой флэш-памяти. Устройство имеет минимальные размеры и допускает «горячее» подключение в разъем USB, после чего распознается как жесткий диск, причем не требует установки драйвера. Объем флэш-дисков может составлять от 32 Мбайт до 1 Гбайт, их распространение сдерживает относительно высокая цена.



Рис. 3.5. Flash-накопитель

Устройства обмена данными

Модем. Устройство, предназначенное для обмена информацией между удаленными компьютерами по каналам связи, принято называть модемом (МОдулятор + ДЕМОдулятор). При этом под каналом связи понимают физические линии (проводные, оптоволоконные, кабельные, радиочастотные), способ их использования (коммутируемые и выделенные) и способ передачи данных (цифровые или аналоговые сигналы). В зависимости от типа канала связи устройства приема-передачи подразделяют на радиомодемы, кабельные модемы и пр. Наиболее широкое применение нашли модемы, ориентированные на подключение к коммутируемым телефонным каналам связи.

Цифровые данные, поступающие в модем из компьютера, преобразуются в нем путем модуляции (по амплитуде, частоте, фазе) в соответствии с избранным стандартом (протоколом) и направляются в телефонную линию. Модем-приемник,

понимающий данный протокол, осуществляет обратное преобразование (демодуляцию) и пересылает восстановленные цифровые данные в свой компьютер. Таким образом обеспечивается удаленная связь между компьютерами и обмен данными между ними.

К основным потребительским параметрам модемов относятся:

- ☐ производительность (бит/с);
- ☐ поддерживаемые протоколы связи и коррекции ошибок;
- ☐ шинный интерфейс, если модем внутренний (ISA или PCI).

От производительности модема зависит объем данных, передаваемых в единицу времени. От поддерживаемых протоколов зависит эффективность взаимодействия данного модема с сопредельными модемами (вероятность того, что они вступят во взаимодействие друг с другом при оптимальных настройках). От шинного интерфейса в настоящее время пока зависит только простота установки и настройки модема (в дальнейшем при общем совершенствовании каналов связи шинный интерфейс начнет оказывать влияние и на производительность).

Практическое занятие

Упражнение 3.1. Подключение оборудования к системному блоку

Работа выполняется под руководством преподавателя (инструктора).

1. Убедитесь в том, что компьютерная система обесточена.
2. Разверните системный блок задней стенкой к себе.
3. По форме разъема клавиатуры установите форм-фактор материнской платы (разъем формата PS/2 — форм-фактор ATX, разъем формата DIN5 — AT).
4. Установите местоположение следующих разъемов:
 - ☐ питания системного блока;
 - ☐ питания монитора;
 - ☐ сигнального кабеля монитора;
 - ☐ клавиатуры;
 - ☐ последовательных портов (два разъема);
 - ☐ параллельного порта.
5. Убедитесь в том, что все разъемы, выведенные на заднюю стенку системного блока, невзаимозаменяемы, то есть каждое базовое устройство подключается одним-единственным способом.
6. При наличии звуковой карты рассмотрите ее разъемы. Установите местоположение следующих разъемов:
 - ☐ подключения головных телефонов;
 - ☐ подключения микрофона;
 - ☐ вывода сигнала на внешний усилитель;

- подключения внешних электромузыкальных инструментов и средств управления компьютерными играми (джойстик, джойпад, геймпад и т. п.).
- 7. Изучите способ подключения мыши. Мышь может подключаться к разъему последовательного порта или к специальному порту PS/2, имеющему разъем круглой формы. Последний способ является более современным и удобным. В этом случае мышь имеет собственный выделенный порт, что исключает возможность ее конфликта с другими устройствами, подключаемыми к последовательным портам. Последние модели могут подключаться к клавиатуре через разъем интерфейса USB.

Упражнение 3.2. Изучение компонентов системного блока

Работа выполняется под руководством преподавателя (инструктора).

1. Убедитесь в том, что компьютерная система обесточена.
2. Установите местоположение блока питания.
3. Установите местоположение материнской платы.
4. Установите характер подключения материнской платы к блоку питания. Для материнских плат в форм-факторе AT подключение питания выполняется двумя разъемами. Обратите внимание на расположение проводников черного цвета — оно важно для правильной стыковки разъемов.
5. Установите местоположение жесткого диска. Установите местоположение его разъема питания. Проследите направление шлейфа проводников, связывающего жесткий диск с материнской платой. Обратите внимание на местоположение проводника, окрашенного в красный цвет (он должен быть расположен рядом с разъемом питания).
6. Установите местоположения дисководов гибких дисков и дисковода CD-ROM. Проследите направление их шлейфов проводников и обратите внимание на положение проводника, окрашенного в красный цвет, относительно разъема питания.
7. Установите местоположение звуковой карты и платы видеоадаптера.
8. При наличии прочих дополнительных устройств задайте инструктору вопросы об их назначении.

Упражнение 3.3. Изучение компонентов материнской платы

Работа выполняется под руководством преподавателя (инструктора).

1. Убедитесь в том, что компьютерная система обесточена.
2. Установите местоположение процессора и изучите организацию его системы охлаждения. По маркировке определите тип процессора и фирму-изготовителя.
3. Установите местоположение разъемов для установки модулей оперативной памяти. Выясните их количество и тип используемых модулей.

4. Установите местоположение слотов для установки плат расширения. Выясните их количество и тип (ISA, VLB, PCI, AGP). Зафиксируйте их различия по форме и цвету.

Разъем шины	Цвет	Размер
ISA	Черный	Длинный
PCI	Белый	Средний
AGP	Коричневый	Короткий
PCI Express X1	Белый или желтый	Короткий
PCI Express X16	Белый или желтый	Длинный

5. Установите местоположение микросхемы ПЗУ. По наклейке на ней определите производителя системы BIOS данного компьютера.
6. Установите местоположение микросхем системного комплекта (чипсета). По маркировке определите тип комплекта и фирму-изготовитель.
7. Заполните отчетные таблицы.

	Изготовитель	Модель
Процессор		
Чипсет		
Система BIOS		-----

Разъемы модулей оперативной памяти		Слоты для установки плат расширения	
Тип	Количество	Тип	Количество
		PCI	
		PCI Express	

Упражнение 3.4. Исследование порядка запуска компьютера

Работа выполняется под руководством преподавателя (инструктора).

- Если монитор вычислительной системы имеет питание, отдельное от системного блока, включите монитор.
- Включите компьютерную систему выключателем системного блока.
- При подаче питания на процессор происходит его обращение к микросхеме ПЗУ и запуск программы, инициализирующей работу компьютера. В этот момент на экране монитора наблюдается сообщение о версии BIOS.
- Для наблюдения сообщений, поступающих от компьютера в процессе запуска, используйте клавишу **Pause/Break**. Она приостанавливает загрузку и дает

возможность внимательно прочесть сообщение. Для продолжения запуска используйте клавишу ENTER.

5. Процедура инициализации запускает процедуру POST, выполняющую само-тестирование базовых устройств (POST — Power-On Self-Test). В этот момент на экране наблюдается сообщение **Memory Test**: и указание объема проверенной памяти компьютера.
6. При отсутствии дефектов в оперативной памяти или клавиатуре происходит обращение к микросхеме CMOS, в которой записаны данные, определяющие состав компьютерной системы и ее настройки. На экране монитора эти данные отображаются в таблице **System Configuration**. Приостановив запуск с помощью клавиши PAUSE/BREAK, изучите таблицу и установите:
 - сколько жестких дисков имеет компьютерная система и каков их объем;
 - имеются ли дисководы гибких дисков и каковы параметры используемых гибких дисков;
 - сколько последовательных и параллельных портов имеется в наличии;
 - к какому типу относятся микросхемы, размещенные в банках памяти.

Продолжите запуск клавишей ENTER.

7. Установив параметры жесткого диска, компьютерная система обращается в его системную область, находит там операционную систему и начинает ее загрузку. Далее работа с компьютером выполняется под управлением операционной системы.
8. Дождавшись окончания запуска операционной системы, выясните у инструктора (преподавателя) порядок завершения работы с компьютером. Приведите компьютер в исходное состояние.

Упражнение 3.5. Настройка компьютерной системы средствами программы SETUP

Работа выполняется под руководством преподавателя (инструктора).

Программа SETUP входит в состав базовой системы ввода-вывода и предназначена для первичной настройки аппаратной конфигурации вычислительной системы. Основная задача настройки — обеспечить возможность автоматического определения состава системы средствами BIOS. Дополнительная задача — оптимизировать настройки и повысить эффективность всей системы в целом.

В большинстве случаев программа SETUP вызывается нажатием клавиши DELETE сразу после включения питания. В отдельных случаях может использоваться иная клавиша или комбинация клавиш — необходимая информация выдается на экран монитора при запуске компьютера.

Неквалифицированное изменение настроек микросхемы CMOS может привести к выходу компьютерной системы из строя. В связи с этим примите следующие меры:

- не вносите никаких изменений в настройки без указания инструктора (преподавателя);

- записью на отдельном листе бумаги четко фиксируйте все параметры до их изменения и после;
- по окончании работы закройте программу SETUP без сохранения внесенных изменений. Перед закрытием программы обратитесь к инструктору (преподавателю) для контроля.

1. Если монитор вычислительной системы имеет питание, отдельное от системного блока, включите монитор.
2. Включите компьютерную систему.
3. При появлении информации на экране нажмите клавишу **DELETE** — произойдет запуск программы **SETUP** и откроется меню, представленное на рис. 3.6.
4. С помощью клавиш управления курсором выберите пункт меню **Standard CMOS Features** (Стандартные настройки микросхемы CMOS).

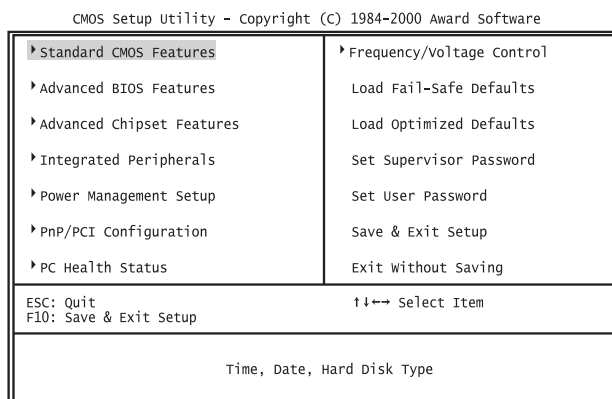


Рис. 3.6. Титульный экран программы SETUP

5. В открывшемся окне проверьте установку системных часов и системного календаря. Выбор настраиваемого параметра выполняется клавишами управления курсором, а изменение параметра — клавишами **PAGE UP/PAGE DOWN**.
6. Вернитесь в предыдущее меню с помощью клавиши **ESC**.
7. Выберите пункт **Advanced BIOS Features** (Настройки параметров BIOS). Нажмите клавишу **ENTER**.
8. В открывшемся окне проверьте, с какого диска начинается запуск компьютера. Последовательность запуска задается в пункте **BOOT SEQUENCE**. С помощью клавиш **PAGE UP** и **PAGE DOWN** просмотрите все возможные для данного компьютера варианты запуска. Особое внимание обратите на вариант запуска, начинающегося с жесткого диска **C:** (он используется при штатной работе), и на вариант запуска, начинающегося с гибкого диска **A:**, — он используется при восстановлении работоспособности компьютера, если загрузка с жесткого диска по каким-то причинам невозможна.
9. Обратите внимание на пункт **Typematic Rate Setting** — если он включен (**Enabled**), то путем настройки BIOS можно управлять настройкой функции автоматиче-

ского повтора символов для клавиатуры (см. раздел 3.1). В этом случае интервал времени до начала повтора определяется установкой параметра **Typematic Rate Delay** (Задержка перед повтором), измеряемого в миллисекундах, а частота повтора определяется установкой параметра **Typematic Rate** (Частота повтора), измеряемого в знаках в секунду.

10. Вернитесь в предыдущее меню нажатием клавиши **ESC**.
11. Завершите работу с программой **SETUP** без сохранения результатов изменения. Для этого нажмите клавишу **ESC** и при получении запроса подтвердите выход без сохранения изменений нажатием клавиши **Y** (**Yes** — Да).

Глава четвертая

Функции операционных систем персональных компьютеров

Операционная система представляет собой комплекс системных и служебных программных средств. С одной стороны, она опирается на базовое программное обеспечение компьютера, входящее в его систему *BIOS* (базовая система ввода-вывода); с другой стороны, она сама является опорой для программного обеспечения более высоких уровней — прикладных и большинства служебных приложений. *Приложениями операционной системы* принято называть программы, предназначенные для работы под управлением данной системы.

Основная функция всех операционных систем — посредническая. Она заключается в обеспечении нескольких видов интерфейса:

- интерфейса между пользователем и программно-аппаратными средствами компьютера (*интерфейс пользователя*);
- интерфейса между программным и аппаратным обеспечением (*аппаратно-программный интерфейс*);
- интерфейса между разными видами программного обеспечения (*программный интерфейс*).

Даже для одной аппаратной платформы, например такой, как *IBM PC*, существует несколько операционных систем. Различия между ними рассматривают в двух категориях: внутренние и внешние. Внутренние различия характеризуются методами реализации основных функций. Внешние различия определяются наличием и доступностью приложений данной системы, необходимых для удовлетворения технических требований, предъявляемых к конкретному рабочему месту.

4.1. Обеспечение интерфейса пользователя

Режимы работы с компьютером

Все операционные системы способны обеспечивать как *пакетный*, так и *диалоговый режим* работы с пользователем. В пакетном режиме операционная система автоматически исполняет заданную последовательность команд. Суть диалогового режима состоит в том, что операционная система находится в ожидании команды пользователя и, получив ее, приступает к исполнению, а исполнив, возвращает отклик и ждет очередной команды. Диалоговый режим работы основан на использовании *прерываний процессора* и *прерываний BIOS* (которые, в свою очередь, также основаны на использовании прерываний процессора). Опираясь на эти *аппаратные прерывания*, операционная система создает свой комплекс *системных прерываний*. Способность операционной системы прервать текущую работу и отреагировать на события, вызванные пользователем с помощью управляющих устройств, воспринимается нами как диалоговый режим работы.

Виды интерфейсов пользователя

Интерфейс командной строки. По реализации интерфейса пользователя различают *неграфические* и *графические операционные системы*. Неграфические операционные системы реализуют *интерфейс командной строки*. Основным устройством управления в данном случае является клавиатура. Управляющие команды вводят в поле командной строки, где их можно и редактировать. Исполнение команды начинается после ее утверждения, например нажатием клавиши **ENTER**. Для компьютеров платформы IBM PC интерфейс командной строки обеспечивается семейством операционных систем под общим названием MS-DOS (версии от MS-DOS 1.0 до MS-DOS 6.2).

Графический интерфейс. Графические операционные системы реализуют более сложный тип интерфейса, в котором в качестве органа управления кроме клавиатуры может использоваться мышь или адекватное устройство позиционирования. Работа с графической операционной системой основана на взаимодействии активных и пассивных экранных элементов управления.

Активные и пассивные элементы управления. В качестве активного элемента управления выступает *указатель мыши* — графический объект, перемещение которого на экране синхронизировано с перемещением мыши.

В качестве пассивных элементов управления выступают графические *элементы управления приложений* (экранные кнопки, значки, переключатели, флажки, раскрывающиеся списки, строки меню и многие другие).

Характер взаимодействия между активными и пассивными элементами управления выбирает сам пользователь. В его распоряжении приемы наведения указателя мыши на элемент управления, щелчки кнопками мыши и другие средства.

4.2. Обеспечение автоматического запуска

Все операционные системы обеспечивают свой автоматический запуск. Для дисковых операционных систем в специальной (системной) области диска создается запись программного кода. Обращение к этому коду выполняют программы, находящиеся в базовой системе ввода-вывода (BIOS). Завершая свою работу, они дают команду на загрузку и исполнение содержимого системной области диска.

Недисковые операционные системы характерны для специализированных вычислительных систем, в частности для компьютеризированных устройств автоматического управления. Математическое обеспечение, содержащееся в микросхемах ПЗУ таких компьютеров, можно условно рассматривать как аналог операционной системы. Автоматический запуск такой системы осуществляется аппаратно. При подаче питания процессор обращается к фиксированному физическому адресу ПЗУ (его можно изменять аппаратно с использованием логических микросхем), с которого начинается запись программы инициализации операционной системы.

4.3. Организация файловой системы

Все современные дисковые операционные системы обеспечивают создание файловой системы, предназначенной для хранения данных на дисках и обеспечения доступа к ним. Принцип организации файловой системы — табличный. Поверхность жесткого диска рассматривается как трехмерная матрица, измерениями которой являются номера *поверхности*, *цилиндра* и *сектора*. Под цилиндром понимается совокупность всех дорожек, принадлежащих разным поверхностям и находящихся на равном удалении от оси вращения. Данные о том, в каком месте диска записан тот или иной файл, хранятся в системной области диска. Формат служебных данных определяется конкретной файловой системой. Нарушение целостности служебных сведений приводит к невозможности воспользоваться данными, записанными на диске. Поэтому к системной области предъявляются особые требования по надежности. Целостность, непротиворечивость и надежность этих данных регулярно контролируется средствами операционной системы.

Наименьшей физической единицей хранения данных является сектор. Размер сектора равен 512 байт. Теоретически возможна самостоятельная адресация для каждого сектора. Но для дисков большого объема такой подход неэффективен, а для некоторых файловых систем — и просто невозможен. В связи с этим группы секторов объединяются в кластеры. *Кластер является наименьшей единицей адресации при обращении к данным.* Размер кластера, в отличие от размера сектора, строго не фиксирован. Обычно он зависит от емкости диска.

Операционные системы MS-DOS, OS/2, Windows 95 и другие используют файловую систему на основе таблиц размещения файлов (FAT-таблицы), состоящих из 16-разрядных полей. Такая файловая система называется FAT16. Она позволяет разместить в FAT-таблицах не более 65 536 записей (2^{16}) о местоположении единиц хранения данных. Для дисков объемом от одного до двух гигабайт длина кластера составляет 32 Кбайт (64 сектора). Это не вполне рациональный расход рабочего пространства, поскольку любой файл (даже очень маленький) полностью оккупирует весь кластер, которому соответствует только одна адресная запись в таблице размещения файлов. Даже если файл достаточно велик и располагается в нескольких кластерах, все равно в его конце образуется некий остаток, нерационально расходуемый целый кластер.

Для жестких дисков, объем которых приближается к двум гигабайтам, потери, связанные с неэффективностью этой файловой системы, весьма значительны и могут составлять от 25 до 40% полной емкости диска, в зависимости от среднего размера хранящихся файлов. С дисками же размером более двух гигабайт файловая система FAT16 вообще работать не может.

Начиная с Windows 98, операционные системы семейства Windows (Windows 98, Windows Me, Windows 2000, Windows XP, Windows Vista, Windows 7) поддерживают более совершенную версию файловой системы на основе FAT-таблиц — FAT32 с 32-разрядными полями в таблице размещения файлов. Для дисков размером до 8 Гбайт эта система обеспечивает размер кластера 4 Кбайт (8 секторов).

Операционные системы Windows NT, Windows XP, Windows Vista, Windows 7 способны поддерживать совершенно другую файловую систему — NTFS. В ней хра-

нение файлов организовано иначе — служебная информация хранится в главной таблице файлов (MFT). В системе NTFS размер кластера не зависит от размера диска, и, потенциально, для очень больших дисков эта система должна работать эффективнее, чем FAT32. Однако с учетом типичных характеристик современных компьютеров можно говорить о том, что в настоящее время эффективность FAT32 и NTFS примерно одинакова.

4.4. Обслуживание файловой структуры

Несмотря на то что данные о местоположении файлов хранятся в табличной структуре, пользователю они представляются в виде иерархической структуры — людям так удобнее, а все необходимые преобразования берет на себя операционная система. К функции обслуживания файловой структуры относятся следующие операции, происходящие под управлением операционной системы:

- ☐ создание файлов и присвоение им имен;
- ☐ создание каталогов (папок) и присвоение им имен;
- ☐ переименование файлов и каталогов (папок);
- ☐ копирование и перемещение файлов между дисками компьютера и между каталогами (папками) одного диска;
- ☐ удаление файлов и каталогов (папок);
- ☐ навигация по файловой структуре с целью доступа к заданному файлу, каталогу (папке);
- ☐ управление атрибутами файлов.

Создание и именование файлов

Файл — это именованная последовательность байтов произвольной длины. Поскольку из этого определения вытекает, что файл может иметь нулевую длину, то фактически создание файла состоит в присвоении ему имени и регистрации его в файловой системе — это одна из функций операционной системы. Даже когда мы создаем файл, работая в какой-то прикладной программе, в общем случае для этой операции привлекаются средства операционной системы.

По способам именования файлов различают «короткое» и «длинное» имя. До появления операционной системы Windows 95 общепринятым способом именования файлов на компьютерах IBM PC было соглашение 8.3. Согласно этому соглашению, принятому в MS-DOS, имя файла состоит из двух частей: собственно *имени* и *расширения имени*. На имя файла отводится 8 символов, а на его расширение — 3 символа. Имя от расширения отделяется точкой. Как имя, так и расширение могут включать только алфавитно-цифровые символы латинского алфавита.

Соглашение 8.3 не является стандартом, и потому в ряде случаев отклонения от правильной формы записи допускаются как операционной системой, так и ее приложениями. Так, например, в большинстве случаев система «не возражает» против использования некоторых специальных символов (восклицательный знак,

символ подчеркивания, дефис, тильда и т. п.), а некоторые версии MS-DOS даже допускают использование в именах файлов символов русского и других алфавитов. Сегодня имена файлов, записанные в соответствии с соглашением 8.3, считаются «короткими».

Основным недостатком «коротких» имен является их низкая содержательность. Далеко не всегда удается выразить несколькими символами характеристику файла, поэтому с появлением операционной системы Windows 95 было введено понятие «длинного» имени. Такое имя может содержать до 256 символов. Этого вполне достаточно для создания содержательных имен файлов. «Длинное» имя может содержать любые символы, кроме девяти специальных: \ / : * ? « < > |. В имени разрешается использовать пробелы и несколько точек. Расширением имени считаются все символы, идущие после последней точки, их может быть и больше трех.

Введение длинных имен потребовало внесения изменений в организацию файловых систем на основе FAT. Появился термин VFAT, обозначающий файловую систему на основе FAT с поддержкой длинных имен. Файловая система NTFS поддерживает длинные имена с самого начала.

Наряду с «длинным» именем операционные системы семейства Windows создают также и короткое имя файла — оно необходимо для возможности работы с данным файлом на рабочих местах с устаревшими операционными системами.

Особенности использования длинных имен. Использование «длинных» имен файлов в операционных системах семейства Windows имеет ряд особенностей.

1. Если «длинное» имя файла включает пробелы, то в служебных операциях его надо заключать в кавычки. Рекомендуется не использовать пробелы, а заменять их символами подчеркивания.
2. В корневой папке диска (на верхнем уровне иерархической файловой структуры) нежелательно хранить файлы с длинными именами. В файловых системах на основе FAT количество единиц хранения в этой папке ограничено. Чем длиннее имена, тем меньше файлов можно разместить в корневой папке.
3. Кроме ограничения на длину имени файла (256 символов) существует гораздо более жесткое ограничение на длину *полного имени файла* (в него входит путь доступа к файлу, начиная от вершины иерархической структуры). Полное имя не может быть длиннее 260 символов.
4. В длинных именах файлов разрешается использовать символы любых алфавитов, в том числе и русского, но если документ готовится для передачи, с заказчиком (потребителем документа) необходимо согласовать возможность воспроизведения файлов с такими именами на его оборудовании.
5. Прописные и строчные буквы в именах не различаются операционной системой. Для нее имена Письмо.txt и письмо.txt соответствуют одному и тому же файлу. Однако отображаются символы разных регистров операционной системой исправно. Если для наглядности желательно использовать прописные буквы, это можно делать.
6. Программисты давно научились использовать расширение имени файла для передачи операционной системе, исполняющей программе или пользователю

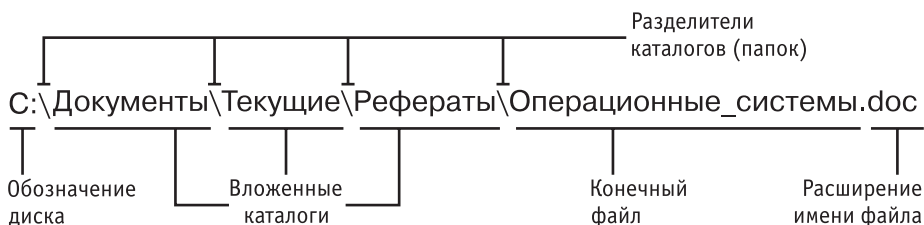
информации о том, к какому типу относятся данные, содержащиеся в файле, и о формате, в котором они записаны. В ранних операционных системах этот факт использовался мало. По существу, операционные системы MS-DOS анализировали только расширения .BAT (пакетные файлы с командами MS-DOS), .EXE, .COM (исполнимые файлы программ) и .SYS (системные файлы конфигурации). В современных операционных системах любое расширение имени файла может нести информацию для операционной системы. Операционные системы семейства Windows имеют средства для регистрации свойств типов файлов по расширению их имени, поэтому во многих случаях выбор расширения имени файла не является частным делом пользователя. Приложения этих систем предлагают выбрать только основную часть имени и указать тип файла, а соответствующее расширение имени приписывают автоматически.

Создание каталогов (папок)

Каталоги (папки) — важные элементы иерархической структуры, необходимые для обеспечения удобного доступа к файлам, если файлов на носителе слишком много. Файлы объединяются в каталоги по любому общему признаку, заданному их создателем (по типу, по принадлежности, по назначению, по времени создания и т. п.). Каталоги низких уровней вкладываются в каталоги более высоких уровней и являются для них *вложенными*. Верхним уровнем вложенности иерархической структуры является *корневой каталог* диска.

Все современные операционные системы позволяют создавать каталоги. Правила присвоения имени каталогу ничем не отличаются от правил присвоения имени файлу, хотя негласно для каталогов не принято задавать расширения имен.

Мы знаем, что в иерархических структурах данных адрес объекта задается *маршрутом (путем доступа)*, ведущим от вершины структуры к объекту. При записи пути доступа к файлу, проходящего через систему вложенных каталогов, все промежуточные каталоги разделяются между собой определенным символом. Во многих операционных системах в качестве такого символа используется «\» (обратная косая черта), например:



Каталоги и папки. До появления операционной системы Windows 95 при описании иерархической файловой структуры использовался введенный выше термин *каталог*. С появлением этой системы был введен новый термин — *папка*. В том, что касается обслуживания файловой структуры носителя данных, эти термины равнозначны: каждому каталогу файлов на диске соответствует одноименная папка

операционной системы. Основное отличие понятий *папка* и *каталог* проявляется не в организации хранения файлов, а в организации хранения объектов иной природы. Так, например, в операционных системах семейства Windows существуют специальные папки, представляющие собой удобные логические структуры, которым не соответствует ни один каталог диска.

Копирование и перемещение файлов

В неграфических операционных системах операции копирования и перемещения файлов выполняются вводом прямой команды в поле командной строки. При этом указывается имя команды, путь доступа к каталогу-источнику и путь доступа к каталогу-приемнику.

В графических операционных системах существуют приемы работы с устройством позиционирования, позволяющие выполнять эти команды наглядными методами.

Удаление файлов и каталогов (папок)

Средства удаления данных не менее важны для операционной системы, чем средства их создания, поскольку ни один носитель данных не обладает бесконечной емкостью. Существует как минимум три режима удаления данных: *удаление*, *уничтожение* и *стирание*, хотя операционные системы обеспечивают только два первых режима (режим надежного стирания данных можно обеспечить лишь специальными программными средствами).

Удаление файлов является временным. В операционных системах семейства Windows оно организовано с помощью специальной папки, которая называется Корзина. При удалении файлов и папок они перемещаются в Корзину. Эта операция происходит на уровне файловой структуры операционной системы (изменяется только путь доступа к файлам). На уровне файловой системы жесткого диска ничего не происходит — файлы остаются в тех же секторах, где и были записаны.

Уничтожение файлов происходит при их удалении в операционной системе MS-DOS или при очистке Корзины в операционных системах семейства Windows. В этом случае файл полностью удаляется из файловой структуры операционной системы, но на уровне файловой системы диска с ним происходят лишь незначительные изменения. В таблице размещения файлов он помечается как удаленный, хотя физически остается там же, где и был. Это сделано для минимизации времени операции. При этом открывается возможность записи новых файлов в кластеры, помеченные как «свободные».

Для справки укажем, что операция *стирания файлов*, выполняемая специальными служебными программами, состоит именно в том, чтобы заполнить якобы свободные кластеры, оставшиеся после уничтоженного файла, случайными данными. Поскольку даже после перезаписи данных их еще можно восстановить специальными аппаратными средствами (путем анализа остаточного магнитного гистерезиса), для надежного стирания файлов требуется провести не менее пяти

актов случайной перезаписи в одни и те же сектора. Эта операция весьма продолжительна, и, поскольку массовому потребителю она не нужна, ее не включают в стандартные функции операционных систем.

Навигация по файловой структуре

Навигация по файловой структуре является одной из наиболее используемых функций операционной системы. Удобство этой операции часто воспринимают как удобство работы с операционной системой. В операционных системах, имеющих интерфейс командной строки, навигацию осуществляют путем ввода команд перехода с диска на диск или из каталога в каталог. В связи с крайним неудобством такой навигации широкое применение нашли специальные служебные программы, называемые *файловыми оболочками*.

Как и операционные системы, файловые оболочки бывают неграфическими и графическими. Наиболее известная неграфическая файловая оболочка для MS-DOS — диспетчер файлов *Norton Commander*. Роль графической файловой оболочки для MS-DOS в свое время исполняли программы Windows 1.0 и Windows 2.0, которые постепенно развились до понятия *операционной среды* (в версиях Windows 3.x) и далее до самостоятельной операционной системы (начиная с Windows 95).

С приемами навигации в современных графических операционных системах мы познакомимся при их изучении.

Управление атрибутами файлов

Кроме имени и расширения имени файла операционная система хранит для каждого файла дату его создания (изменения) и несколько флаговых величин, называемых *атрибутами файла*. Атрибуты — это дополнительные параметры, определяющие свойства файлов. Операционная система позволяет их контролировать и изменять; состояние атрибутов учитывается при проведении автоматических операций с файлами.

Основных атрибутов четыре:

- ☐ Только для чтения (Read only);
- ☐ Скрытый (Hidden);
- ☐ Системный (System);
- ☐ Архивный (Archive).

Атрибут Только для чтения ограничивает возможности работы с файлом. Его установка означает, что файл не предназначен для внесения изменений.

Атрибут Скрытый сигнализирует операционной системе о том, что данный файл не следует отображать на экране при проведении файловых операций. Это мера защиты против случайного (умышленного или неумышленного) повреждения файла.

Атрибутом Системный помечаются файлы, обладающие важными функциями для работы самой операционной системы. Его отличительная особенность в том, что средствами операционной системы его изменить нельзя. Как правило, большинство файлов, имеющих установленный атрибут Системный, имеют также и установленный атрибут Скрытый.

Атрибут **Архивный** в прошлом использовался для работы программ резервного копирования. Предполагалось, что любая программа, изменяющая файл, должна автоматически устанавливать этот атрибут, а средство резервного копирования должно его сбрасывать. Таким образом, очередному резервному копированию подлежали только те файлы, у которых этот атрибут был установлен. Современные программы резервного копирования используют другие средства для установления факта изменения файла, и данный атрибут во внимание не принимается, а его изменение вручную средствами операционной системы не имеет практического значения.

4.5. Управление установкой, исполнением и удалением приложений

Понятие многозадачности

Работа с приложениями составляет наиболее важную часть работы операционной системы. Это очевидно, если вспомнить, что основная функция операционной системы состоит в обеспечении интерфейса приложений с аппаратными и программными средствами вычислительной системы, а также с пользователем. С точки зрения управления исполнением приложений различают *однозадачные* и *многозадачные* операционные системы.

Однозадачные операционные системы (например, MS-DOS) передают все ресурсы вычислительной системы одному исполняемому приложению и не допускают ни параллельного выполнения другого приложения (*полная многозадачность*), ни его приостановки и запуска другого приложения (*вытесняющая многозадачность*). В то же время параллельно с однозадачными операционными системами возможна работа специальных программ, называемых *резидентными*. Такие программы не опираются на операционную систему, а непосредственно работают с процессором, используя его систему прерываний.

Большинство современных графических операционных систем — *многозадачные*. Они управляют распределением ресурсов вычислительной системы между задачами и обеспечивают:

- ☐ возможность одновременной или поочередной работы нескольких приложений;
- ☐ возможность обмена данными между приложениями;
- ☐ возможность совместного использования программных, аппаратных, сетевых и прочих ресурсов вычислительной системы несколькими приложениями.

Вопросы надежности

От того, как операционная система управляет работой приложений, во многом зависит надежность всей вычислительной системы. Операционная система должна предоставлять возможность прерывания работы приложений по желанию пользователя и снятия сбойной задачи без ущерба для работы других приложений. При этом требование надежности операционной системы может входить в противоречие с требованием ее универсальности.

У операционных систем семейства *Windows* последних поколений долгое время наблюдались две линии развития. В линию универсальных операционных систем входили Windows 95, Windows 98 и Windows Me. Эти системы могут испытывать общесистемные сбои из-за работы с приложениями, недостаточно четко соблюдающими спецификацию операционной системы. Операционные системы Windows NT и Windows 2000 обладают повышенной устойчивостью и не выходят из строя при сбое приложений. Однако они менее универсальны, и, соответственно, парк доступных приложений для них ограничен.

Попытка объединить достоинства обеих линий сделана в операционной системе Windows XP. Эта система сегодня активно распространяется по массовым многоцелевым вычислительным системам, но постепенно проникает и на специализированные рабочие места, где требуется повышенная надежность при ограничении круга используемых программ. Windows Vista и Windows 7 продолжают идеологию Windows XP. Особенностью современных операционных систем, таких как Windows 7, является наличие различных выпусков (версий) операционной системы. Различные выпуски предназначены для различных категорий пользователей, имеют разную стоимость и различные возможности. Например, Windows 7 Домашняя расширенная предназначена для использования дома. Windows 7 Профессиональная — для использования в организациях и на предприятиях, а также — для профессиональных пользователей.

Установка приложений

Для правильной работы приложений на компьютере они должны пройти операцию, называемую *установкой*. Необходимость в установке связана с тем, что разработчики программного обеспечения не могут заранее предвидеть особенности аппаратной и программной конфигурации вычислительной системы, на которой предстоит работать их программам. Таким образом, *дистрибутивный комплект (установочный пакет)* программного обеспечения, как правило, представляет собой не законченный программный продукт, а полуфабрикат, из которого в процессе установки на компьютере формируется полноценное рабочее приложение. При этом осуществляется привязка приложения к существующей аппаратно-программной среде и его настройка на работу именно в этой среде.

Устаревшие операционные системы (например, MS-DOS) не имеют средств для управления установкой приложений. Единственное средство, которое они предоставляют, — возможность запуска устанавливающей программы, прилагаемой к дистрибутивному комплекту. Такая установка отличается крайней простотой, но и невысокой надежностью, поскольку правильность привязки приложения к окружающей программно-аппаратной среде зависит от того, насколько разработчик устанавливающей программы сумел заранее предусмотреть возможные варианты конфигурации вычислительной системы конкретного пользователя.

Современные графические операционные системы берут на себя управление установкой приложений. Они управляют распределением ресурсов вычислительной системы между приложениями, обеспечивают доступ устанавливаемых приложений к драйверам устройств вычислительной системы, формируют общие

ресурсы, которые могут использоваться разными приложениями, выполняют регистрацию установленных приложений и выделенных им ресурсов.

Удаление приложений

Процесс удаления приложений, как и процесс установки, имеет свои особенности и может происходить под управлением вычислительной системы. В таких операционных системах, где каждое приложение самообеспечено собственными ресурсами (например, в MS-DOS), его удаление не требует специального вмешательства операционной системы. Для этого достаточно удалить каталог, в котором размещается приложение, со всем его содержимым.

В операционных системах, реализующих принцип совместного использования ресурсов (например, в системах семейства Windows), процесс удаления приложений имеет особенности. Нельзя допустить, чтобы при удалении одного приложения были удалены ресурсы, на которые опираются другие приложения, даже если эти ресурсы были когда-то установлены вместе с удаляемым приложением. В связи с этим удаление приложений происходит под строгим контролем операционной системы. Полнота удаления и надежность последующего функционирования операционной системы и оставшихся приложений во многом зависят от корректности установки и регистрации приложений в реестре операционной системы.

4.6. Взаимодействие с аппаратным обеспечением

Средства аппаратного обеспечения вычислительной техники отличаются гигантским многообразием. Существуют сотни различных моделей видеоадаптеров, звуковых карт, мониторов, принтеров, сканеров и прочего оборудования. Ни один разработчик программного обеспечения не в состоянии предусмотреть все варианты взаимодействия своей программы, например, с печатающим устройством.

Гибкость аппаратных и программных конфигураций вычислительных систем поддерживается за счет того, что каждый разработчик оборудования прикладывает к нему специальные программные средства управления — драйверы. Драйверы имеют *точки входа* для взаимодействия с прикладными программами, а диспетчеризация обращений прикладных программ к драйверам устройств — это одна из функций операционной системы. Строго говоря, выпуская устройство, например модем, его разработчик прикладывает к нему несколько драйверов, предназначенных для основных операционных систем, как-то: MS-DOS, Windows XP, Windows Vista, Windows 7, Linux и т. п.

В операционных системах MS-DOS драйверы устройств загружаются как *резидентные программы*, напрямую работающие с процессором и другими устройствами материнской платы. Здесь участие операционной системы сводится лишь к тому, чтобы предоставить пользователю возможность загрузки драйвера — далее он сам перехватывает прерывания, используемые для обращения к устройству, и управляет его взаимодействием с вызывающей программой. Загрузка драйверов устройств может быть ручной или автоматической. При ручной загрузке после первоначальной загрузки компьютера пользователь сам выдает команды на загрузку

драйверов. В автоматическом режиме команды на загрузку и настройку драйверов включаются в состав файлов, автоматически читаемых при загрузке компьютера. В MS-DOS такие файлы называются *файлами конфигурации*; их всего два — это файлы `autoexec.bat` и `config.sys`. В них прежде всего включают команды загрузки драйвера мыши, дисководов CD-ROM, звуковой карты, расширенной памяти (оперативная память, лежащая за пределами одного мегабайта, рассматривается в MS-DOS как дополнительное устройство и требует специального драйвера), а также прочих устройств.

В операционных системах семейства Windows операционная система берет на себя все функции по установке драйверов устройств и передаче им управления от приложений. Во многих случаях операционная система даже не нуждается в драйверах, полученных от разработчика устройства, а использует драйверы из собственной базы данных.

Наиболее современные операционные системы позволяют управлять не только установкой и регистрацией программных драйверов устройств, но и процессом аппаратно-логического подключения. Каждое подключенное устройство может использовать до трех аппаратных ресурсов устройств материнской платы: *адресов внешних портов процессора, прерываний процессора и каналов прямого доступа к памяти*.

При некоторых способах подключения устройства к материнской плате (например, через шину PCI) есть техническая возможность организовать между ним и материнской платой обратную связь. Это позволяет операционной системе анализировать требования устройств о выделении им ресурсов и гибко реагировать на них, исключая захват одних и тех же ресурсов разными устройствами. Такой принцип динамического распределения ресурсов операционной системой получил название `plug-and-play`, а устройства, удовлетворяющие этому принципу, называются *самоустанавливающимися*.

Устройства, подключаемые по устаревшим шинам, не являются самоустанавливающимися. В этом случае операционная система не может выделять им ресурсы динамически, но тем не менее при распределении ресурсов для самоустанавливающихся устройств она учитывает ресурсы, захваченные ими.

4.7. Обслуживание компьютера

Предоставление основных средств обслуживания компьютера — одна из функций операционной системы. Обычно она решается внешним образом — включением в базовый состав операционной системы первоочередных служебных приложений.

Средства проверки дисков

Надежность работы дисков (особенно жесткого диска) определяет не только надежность работы компьютера в целом, но и безопасность хранения данных, ценность которых может намного превышать стоимость самого компьютера. Поэтому наличие средств для проверки дисков является обязательным требованием к любой операционной системе.

Средства проверки принято рассматривать в двух категориях: средства логической проверки, то есть проверки целостности файловой структуры, и средства физической диагностики поверхности. Логические ошибки, как правило, устраняются средствами самой операционной системы, а физические дефекты поверхности только локализуются — операционная система принимает во внимание факт повреждения магнитного слоя в определенных секторах и исключает их из активной работы.

Возможность возникновения логических ошибок зависит от типа файловой системы. Например, схема организации работы в системе NTFS вообще исключает возникновение внутренних несоответствий в логической структуре, если не принимать во внимание возможность физического сбоя в процессе записи.

В системе на основе FAT логические ошибки файловой структуры имеют два характерных проявления: это *потерянные кластеры* или *общие кластеры*. Потерянные кластеры образуются в результате неправильного (или аварийного) завершения работы с компьютером. Так, например, ни в одной операционной системе нельзя выключать компьютер, если на нем запущены приложения, осуществляющие обмен информацией с дисками. Кроме того, в операционных системах Windows также нельзя выключать компьютер, если не исполнена специальная процедура завершения работы с операционной системой. Механизм образования потерянных кластеров выглядит так:

- ❑ во время работы с файлом приложение манипулирует с кластерами, занимая или освобождая их, и регистрирует сведения об этом в FAT-таблице, но не записывает полные сведения о файле в каталог;
- ❑ если при завершении работы с приложением происходит сохранение результатов деятельности, оно вносит окончательные изменения в FAT-таблицы и регистрирует данные, записанные в кластерах, как файл в каталоге;
- ❑ если при завершении работы с приложением файл уничтожается, информация не фиксируется в каталоге, а использованные кластеры освобождаются;
- ❑ если компьютер выключается до завершения работы с приложением, кластеры остаются помеченными как «занятые», но ссылки на них в каталоге не создаются, так что согласно данным FAT-таблицы этим кластерам не соответствует ни один файл.

Ошибка, связанная с *потерянными кластерами*, легко парируется средствами операционной системы. При этом можно либо полностью освободить данные кластеры, либо превратить их в полноценные файлы, которые можно просмотреть в поисках ценной информации, утраченной во время сбоя.

Ошибка, проявляющаяся как *общие кластеры*, характеризуется тем, что, согласно данным FAT-таблиц, два или более файлов претендуют на то, что их данные находятся в одном и том же месте диска. При нормальной работе такой ситуации быть не может, и это свидетельствует об ошибке в FAT-таблицах. Причиной появления общих кластеров может стать самопроизвольное изменение данных в FAT-таблицах или некорректное восстановление ранее удаленных данных с помощью внесистемных средств. Некорректность может быть обусловлена нарушением порядка операций восстановления данных или неадекватностью средств восстановления

данных (например, использованием средств MS-DOS для восстановления файлов, записанных средствами Windows).

Ошибка, связанная с общими кластерами, парируется повторной записью обоих конфликтующих файлов. Один из них обязательно испорчен и подлежит последующему удалению, но велика вероятность того, что испорчены оба файла.

Дополнительно к вышеуказанным логическим ошибкам операционные системы семейства Windows определяют логические ошибки, связанные с некорректной записью даты создания файла и с представлением «короткого» имени файла для заданного «длинного» имени.

В операционной системе Windows 7 проверка дисков, содержащих системную или служебную информацию, рассматривается как потенциально опасная операция, способная поставить дальнейшую работу компьютера под угрозу. В этом случае проверка не выполняется немедленно, а назначается на время очередной перезагрузки системы. Такая же проверка системных дисков обычно производится и в случае аварийного отключения или аварийной перезагрузки компьютера.

Средства «сжатия» дисков

Некоторые операционные системы предоставляют служебные средства для программного «сжатия» дисков путем записи данных на диск в уплотненном виде посредством специального драйвера (резидентного для MS-DOS или работающего в фоновом режиме для Windows). Механизм работы этих средств будет рассмотрен в главе 14.

Средства управления виртуальной памятью

Ранние операционные системы ограничивали возможность использования приложений по объему необходимой для их работы оперативной памяти. Так, например, без специальных драйверов (*менеджеров оперативной памяти*) операционные системы MS-DOS ограничивали предельный размер исполняемых программ величиной около 640 Кбайт.

Современные операционные системы не только обеспечивают непосредственный доступ ко всему полю оперативной памяти, установленной в компьютере, но и позволяют ее расширить за счет создания так называемой *виртуальной памяти* на жестком диске. Виртуальная память реализуется в виде так называемого *файла подкачки*. В случае недостаточности оперативной памяти для работы приложения часть ее временно опорожняется с сохранением образа на жестком диске. В процессе работы приложений происходит многократный обмен между основной установленной оперативной памятью и файлом подкачки. Поскольку электронные операции в оперативной памяти происходят намного быстрее, чем механические операции взаимодействия с диском, увеличение размера оперативной памяти компьютера всегда благоприятно сказывается на ускорении операций и повышении производительности всей вычислительной системы.

Операционная система не только берет на себя весь необходимый обмен данными между ОЗУ и диском, но и позволяет в определенной степени управлять размером файла подкачки вручную.

Средства кэширования дисков

Поскольку, как уже было отмечено, взаимодействие процессора с дисками компьютера происходит намного медленнее операций обмена с оперативной памятью, операционная система принимает специальные меры по сохранению части прочитанных с диска данных в оперативной памяти. В случае если по ходу работы процессору вновь потребуется обратиться к ранее считанным данным или программному коду, он может найти их в специальной области ОЗУ, называемой *дисковым кэшем*. В ранних операционных системах функции кэширования диска возлагались на специальное внешнее программное средство, подключаемое через файлы конфигурации. В современных операционных системах эту функцию включают в ядро системы, и она работает автоматически, без участия пользователя, хотя определенная возможность настройки размера кэша за ним сохраняется.

Средства резервного копирования данных

Если на компьютере выполняется практическая работа, объем ценных (а зачастую и уникальных) данных нарастает с каждым днем. Ценность данных, размещенных на компьютере, принято измерять совокупностью затрат, которые может понести владелец в случае их утраты. Важным средством защиты данных является регулярное резервное копирование на внешний носитель. В связи с особой важностью этой задачи операционные системы обычно содержат базовые средства для выполнения резервного копирования.

4.8. Прочие функции операционных систем

Кроме основных (базовых) функций операционные системы могут предоставлять различные дополнительные функции. Конкретный выбор операционной системы определяется совокупностью предоставляемых функций и конкретными требованиями к рабочему месту.

Прочие функции операционных систем могут включать следующие:

- ☐ возможность поддерживать функционирование локальной компьютерной сети без специального программного обеспечения;
- ☐ обеспечение доступа к основным службам Интернета средствами, интегрированными в состав операционной системы;
- ☐ возможность создания системными средствами сервера Интернета, его обслуживание и управление, в том числе дистанционное посредством удаленного соединения;
- ☐ наличие средств защиты данных от несанкционированного доступа, просмотра и внесения изменений;
- ☐ возможность оформления рабочей среды операционной системы, в том числе и средствами, относящимися к категории мультимедиа;
- ☐ возможность обеспечения комфортной поочередной работы различных пользователей на одном персональном компьютере с сохранением персональных

настроек рабочей среды каждого из них и ограничением доступа к конфиденциальной информации;

- ☐ возможность автоматического исполнения операций по обслуживанию компьютера и операционной системы в соответствии с заданным расписанием или под управлением удаленного сервера;
- ☐ возможность работы с компьютером для лиц, имеющих физические недостатки, связанные с органами зрения, слуха и другими.

Кроме всего вышеперечисленного, современные операционные системы могут включать минимальный набор прикладного программного обеспечения, которое можно использовать для исполнения простейших практических задач:

- ☐ чтение, редактирование и печать текстовых документов;
- ☐ создание и редактирование простейших рисунков;
- ☐ выполнение арифметических и математических расчетов;
- ☐ ведение дневников и служебных блокнотов;
- ☐ создание, передача и прием сообщений электронной почты;
- ☐ создание и редактирование факсимильных сообщений;
- ☐ воспроизведение и редактирование звукозаписи;
- ☐ воспроизведение видеозаписи;
- ☐ разработка и воспроизведение комплексных электронных документов, включающих текст, графику, звукозапись и видеозапись.

Этим возможности операционных систем не исчерпываются. По мере развития аппаратных средств вычислительной техники и средств связи функции операционных систем непрерывно расширяются, а средства их исполнения совершенствуются.

Подведение итогов

Основные достоинства персональной вычислительной техники проявляются в *диалоговом режиме* работы с пользователем. Диалоговый режим отличается от *пакетного* тем, что в ходе работы процессор регулярно приостанавливает выполнение текущих задач и обращается к другим устройствам и программам, проверяя их состояние. Если пользователь использует какое-либо средство управления или извне поступает управляющий сигнал, процессор устанавливает этот факт и реагирует на него переходом на исполнение другой программы. Несмотря на то что в любой момент времени процессор работает по жестко заданным программам, динамичное переключение между ними создает впечатление гибкого управления работой компьютера.

Организацией работы процессора в таком режиме ведает относительно небольшая группа системных программ. Она образует *ядро операционной системы*. Дополнительно к ядру операционная система обладает средствами для:

- ☐ управления пользовательским интерфейсом компьютера;
- ☐ управления аппаратно-программными интерфейсами компьютера;

- ☐ обслуживания файловой системы;
- ☐ управления распределением оперативной памяти между процессами;
- ☐ установки программ и управления их работой;
- ☐ обеспечения надежности и устойчивости работы оборудования и программ.

Чем шире функциональные возможности операционной системы, тем большие требования она предъявляет к техническим ресурсам компьютерной системы, но тем проще работа с компьютером с точки зрения пользователя. Вопрос ресурсной обеспеченности компьютера, универсальности операционной системы, ее надежности, обеспеченности прикладными программами и драйверами устройств, а также простоты и удобства ее использования — это сложный вопрос баланса, который может по-разному решаться на каждом рабочем месте в зависимости от конкретных задач.

Программы, которые работают под управлением операционных систем, называются их *приложениями*. В графических операционных системах принцип управления приложениями состоит во взаимодействии активных и пассивных элементов управления. Активный элемент управления — *указатель мыши* (его предоставляет операционная система). Пассивные элементы управления — *графические кнопки, поля, флажки, переключатели, меню, списки* и пр. Их предоставляют конкретные приложения. В момент взаимодействия активного и пассивного элементов управления пользователь выдает управляющие сигналы с помощью органов управления графического манипулятора.

В неграфических операционных системах управление приложениями ограничено и осуществляется путем ручного ввода текстовых команд в поле командной строки. Органом управления в данном случае является клавиатура.

Вопросы для самоконтроля

1. Что такое операционная система?
2. Перечислите основные функции операционной системы.
3. Расскажите о видах интерфейса пользователя, применяемых в разных операционных системах.
4. Опишите организацию хранения файлов на дисках компьютера.
5. Перечислите функции операционной системы по обслуживанию файловой структуры.
6. Объясните правила, по которым формируются короткое имя файла и длинное имя файла.
7. В чем заключается операция установки приложения?
8. В чем опасность операции удаления приложения?

Глава пятая

Основы работы с операционной системой Windows 7

В предыдущей главе мы рассмотрели функции ряда операционных систем и требования к ним. Надо сказать, что многие из этих требований являются противоречивыми. Например, соотношение требований безотказности и совместимости с приложениями иных систем — это вопрос баланса. Соотношение требований безопасности и простоты обеспечения сетевых функций — это тоже вопрос баланса. На каждом конкретном рабочем месте эти вопросы решаются индивидуально.

В этом смысле сегодня особое место занимает операционная система Windows 7. Она обладает наибольшей универсальностью, имеет самое широкое распространение и, соответственно, получает особую поддержку со стороны производителей аппаратного и программного обеспечения. Для компьютера, работающего в этой системе, наиболее просто подобрать прикладные программы и драйверы устройств.

Почти все, что здесь сказано об операционной системе Windows 7, можно отнести и к другим операционным системам семейства Windows. В том, что касается приемов и методов работы, они в значительной степени совпадают.

5.1. Основные объекты и приемы управления Windows

Windows 7 является графической операционной системой для компьютеров платформы IBM PC. Ее основные средства управления — графический манипулятор (мышь или иной аналогичный) и клавиатура. Система предназначена для управления автономным компьютером, но также содержит все необходимое для создания небольшой локальной компьютерной сети (*одноранговой сети*) и имеет средства для интеграции компьютера во всемирную сеть (Интернет).

Рабочий стол Windows 7

Стартовый экран Windows 7 представляет собой системный объект, называемый *Рабочим столом*. Практически экран Windows 7 является Рабочим столом. Однако существуют видеоадаптеры, позволяющие создать Рабочий стол, размер которого больше, чем видимый размер экрана. Кроме того, Windows 7 имеет штатные средства, позволяющие разместить Рабочий стол на нескольких экранах, если к компьютеру подключено несколько мониторов.

Рабочий стол — это *графическая среда*, на которой отображаются *объекты* Windows и *элементы управления* Windows. Все, с чем мы имеем дело, работая с компьютером в данной системе, можно отнести либо к *объектам*, либо к *элементам управления*. В исходном состоянии на Рабочем столе можно наблюдать несколько экранных значков и Панель задач (рис. 5.1). Значки — это графическое представление объектов Windows, а Панель задач — один из основных элементов управления.

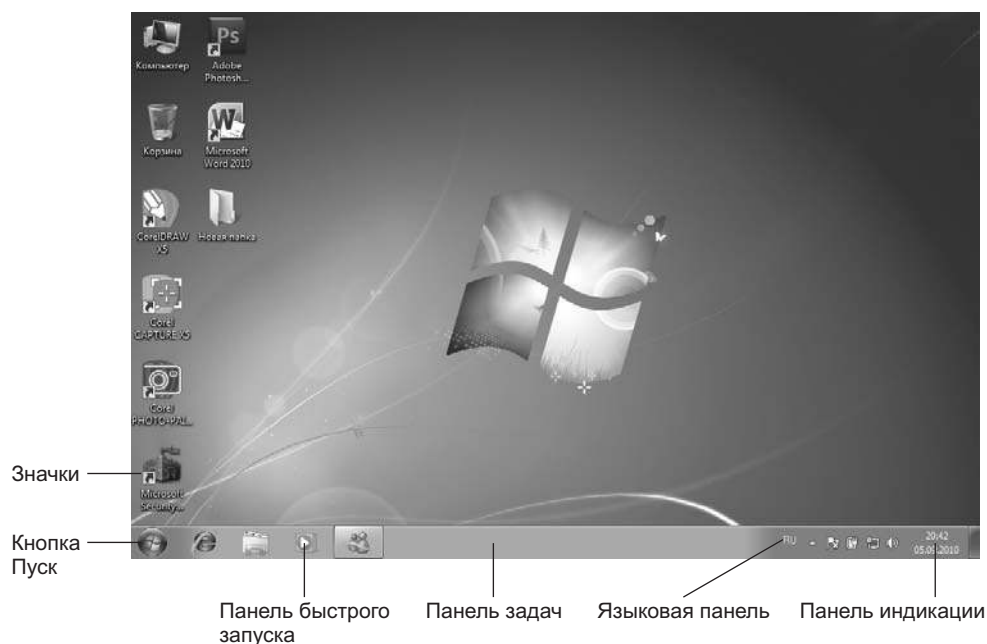


Рис. 5.1. Рабочий стол Windows 7

Управление Windows 7

В Windows 7 большую часть команд можно выполнять с помощью мыши. С мышью связан активный элемент управления — *указатель мыши*. При перемещении мыши по плоской поверхности указатель перемещается по Рабочему столу и его можно *позиционировать* на значках объектов или на пассивных элементах управления приложений.

Основными приемами управления с помощью мыши являются:

- ☐ *щелчок* — быстрое нажатие и отпускание левой кнопки мыши;
- ☐ *двойной щелчок* — два щелчка, выполненные с малым интервалом времени между ними;
- ☐ *щелчок правой кнопкой* — то же, что и *щелчок*, но с использованием правой кнопки;
- ☐ *перетаскивание* (drag-and-drop) — выполняется путем перемещения мыши при нажатой левой кнопке (обычно сопровождается перемещением экранного объекта, на котором установлен указатель);
- ☐ *протягивание мыши* (click-and-drag) — выполняется, как и перетаскивание, но при этом происходит не перемещение экранного объекта, а изменение его формы;
- ☐ *специальное перетаскивание* — выполняется, как и перетаскивание, но при нажатой правой кнопке мыши, а не левой;

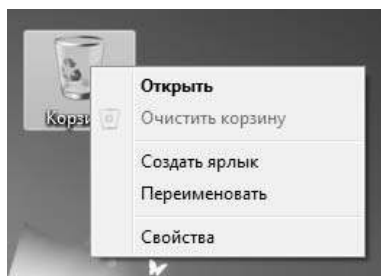
- ❑ *зависание* — наведение указателя мыши на значок объекта или на элемент управления и задержка его на некоторое время (при этом обычно на экране появляется *всплывающая подсказка*, кратко характеризующая свойства объекта).

Значки и ярлыки объектов

Создание ярлыков объектов — это одна из функций приема специального перетаскивания, но нам надо пояснить, что же такое *ярлык*. Рассмотрим это понятие на примере Корзины.

Корзина — специальный объект Windows, выполняющий функции *контейнера*. Она служит для временного хранения удаляемых объектов. Если какой-то документ или программа стали не нужны, их можно удалить, но при этом они не удаляются безвозвратно, а откладываются в Корзину, из которой их впоследствии можно восстановить.

Откройте окно Компьютер и попробуйте перетащить в него значок Корзины. Это не получится, поскольку Корзина — *реквизитный значок* Рабочего стола. Невозможность перетаскивания отображается специальным указателем мыши.



Теперь попробуйте перетащить значок Корзины в окно Документы. Обратите внимание на то, что возле указателя мыши появляется небольшая стрелочка, которая показывает, что при отпускании кнопки мыши будет создан *ярлык* — копия значка Корзина со стрелкой в левом нижнем углу. Ярлыком можно пользоваться точно так же, как обычно пользуются значками.

Значок является *графическим представлением объекта*. То, что мы делаем со значком, мы на самом деле делаем с объектом. Например, удаление значка приводит к удалению объекта; копирование значка приводит к копированию объекта и т. д. Ярлык же является только *указателем* на объект. Удаление ярлыка приводит к удалению указателя, но не объекта; копирование ярлыка приводит к копированию указателя, но не объекта.

Для пользователя приемы работы с ярлыками ничем не отличаются от приемов работы со значками. Точно так же можно запускать программы двойным щелчком на их ярлыках, так же можно и открывать документы. Зато ярлыки позволяют экономить место на жестком диске.

Если объект (например, файл с текстовым документом) имеет большой размер, то его многократное копирование в различные окна папок привело бы фактически к появлению новых объектов (копий файла). При этом многократно увеличился

бы расход рабочего пространства на жестком диске, а у пользователя появились бы сложнейшие заботы по синхронизации содержимого этих копий (при редактировании одной копии ее изменения без специальных мер никак не отразятся на содержимом других копий).

С другой стороны, ярлык является лишь указателем, он занимает ничтожно мало места, и его размножение позволяет обеспечить удобный доступ к связанному с ним объекту из разных мест операционной системы. При этом расход рабочего пространства на жестком диске ничтожен и нет проблем с синхронизацией данных. Из какой бы папки ни открывался документ щелчком на его ярлыке, редактированию всегда подвергается только один связанный с ним объект.

5.2. Файлы и папки Windows

Способ хранения файлов на дисках компьютера называется *файловой системой*. Иерархическая структура, в виде которой операционная система отображает файлы и папки диска, называется *файловой структурой*. Как все дисковые операционные системы, Windows 7 предоставляет средства для управления этой структурой.

Просмотр папок Windows

Откройте окно Компьютер и найдите в нем значок жесткого диска C:. Щелкните на нем дважды, и на экране откроется новое окно, в котором представлены значки объектов, присутствующих на жестком диске. Обратите внимание на значки, представляющие папки, и значки, представляющие файлы. Двойной щелчок на значке любой папки открывает ее окно и позволяет ознакомиться с содержимым. Так можно погружаться в глубь структуры папок до последнего уровня вложения. В соответствующем окне будут представлены только значки файлов.

Окно папки

Окно папки — это *контейнер*, содержимое которого графически отображает содержимое папки. Любую папку Windows можно открыть в своем окне. Количество одновременно открытых окон может быть достаточно большим — это зависит от параметров конкретного компьютера. Окна — одни из самых важных объектов Windows. Абсолютно все операции, которые мы делаем, работая с компьютером, происходят либо на Рабочем столе, либо в каком-либо окне.

Окна папок — не единственный тип окон в Windows. По наличию однородных элементов управления и оформления можно выделить и другие типы окон: *диалоговые окна*, *окна справочной системы* и *рабочие окна приложений*, а внутри окон многих приложений могут существовать отдельные окна документов (если приложение позволяет работать с несколькими документами одновременно).

Если подходить к терминологии с академической строгостью, то за каждым открытым окном скрывается некое работающее приложение (принято говорить *процесс*) и все окна можно было бы назвать окнами приложений (*окнами процессов*), но в учебных целях их лучше все-таки рассматривать порознь.

Структура окна

На рис. 5.2 представлено окно папки \Windows. Такая папка обычно имеется на всех компьютерах, работающих в любой операционной системе семейства Windows. Окно папки содержит следующие элементы.



Рис. 5.2. Окно папки Windows

Строка заголовка. За эту строку выполняется перетаскивание папки на Рабочем столе с помощью мыши.

Служебная область. Находится в левом верхнем углу любого окна папки. При щелчке на этом значке открывается меню, называемое *служебным*. Команды, представленные в данном меню, позволяют управлять размером и расположением окна на Рабочем столе — они могут быть полезны, если мышь не работает.

Кнопки управления размером. Эти кнопки дублируют основные команды служебного меню. В операционной системе Windows 7 исключительно много дублирования. Большинство операций можно выполнить многими различными способами. Каждый пользуется теми приемами, которые ему удобны. Кнопок управления размером три: *закрывающая*, *сворачивающая*, *разворачивающая*.

Щелчок на закрывающей кнопке закрывает окно полностью (и прекращает процесс). Щелчок на сворачивающей кнопке приводит к тому, что окно сворачивается до размера кнопки, которая находится на Панели задач (при этом процесс,

связанный с окном, не прекращается). В любой момент окно можно восстановить щелчком на кнопке Панели задач.

Щелчок на разворачивающей кнопке разворачивает окно на полный экран. При этом работать с ним удобно, но доступ к прочим окнам затрудняется. В развернутом окне разворачивающая кнопка сменяется восстанавливающей, с помощью которой можно восстановить исходный размер окна.

Строка меню. В Windows 7 по умолчанию скрыта. Для ее отображения необходимо нажать на клавишу Alt на клавиатуре. Важнейшие команды меню продублированы на *Панели инструментов*. При щелчке на каждом из пунктов этого меню открывается «ниспадающее» меню, пункты которого позволяют проводить операции с содержимым окна или с окном в целом.

Строка поиска. Позволяет осуществлять быстрый поиск в текущей папке и во вложенных в нее папках по ключевым словам, которые нужно задать в этой строке.

Использование команд, доступных через строку меню, в большинстве случаев не самый эффективный прием работы в Windows (есть и более удобные элементы и средства управления), но зато строка меню гарантированно предоставляет *доступ ко всем командам*, которые можно выполнить в данном окне. Это удобно, если неизвестно, где находится нужный элемент управления. Поэтому при изучении работы с новым приложением в первое время принято пользоваться командами строки меню и лишь потом переходить к использованию других средств управления, постепенно повышая эффективность работы.

Панель инструментов. Содержит командные кнопки для выполнения наиболее часто встречающихся операций. В работе удобнее, чем строка меню, но ограничена по количеству команд. В окнах современных приложений панель инструментов часто бывает настраиваемой. Пользователь сам может разместить на ней те командные кнопки, которыми он пользуется чаще всего.

Адресная строка. В ней указан путь доступа к текущей папке, что удобно для ориентации в файловой структуре. Адресная строка позволяет выполнить быстрый переход к другим разделам файловой структуры с помощью раскрывающихся кнопок.

Кнопки «Вперед» и «Назад». Позволяют передвигаться по файловой структуре с учетом предыдущих переходов.

Рабочая область. В ней отображаются значки объектов, хранящихся в папке, причем способом отображения можно управлять (см. ниже). В окнах приложений в рабочей области размещаются окна документов и рабочие панели.

Область переходов. Используется для навигации по файловой системе компьютера и по логически выделенным объектам. Использует древовидную структуру представления информации, на верхнем уровне которой находятся наиболее важные объекты. Эта область по традициям, заложенным более ранними версиями Windows, нередко называют «Проводником». В Windows 7 она присутствует в окнах папок по умолчанию, без дополнительных настроек.

Полосы прокрутки. Если количество объектов слишком велико (или размер окна слишком мал), по правому и нижнему краям рабочей области могут отображаться

полосы прокрутки, с помощью которых можно «прокручивать» содержимое папки в рабочей области.

Полоса прокрутки имеет движок и две концевые кнопки. Прокрутку выполняют тремя способами:

- ☐ щелчком на одной из концевых кнопок;
- ☐ перетаскиванием движка;
- ☐ щелчком на полосе прокрутке выше или ниже движка.

Область сведений. Здесь выводится дополнительная, часто немаловажная информация. Так, например, если среди объектов, представленных в окне, есть скрытые или системные, они могут не отображаться при просмотре, но в строке состояния об их наличии имеется специальная запись.

5.3. Операции с файловой структурой

К основным операциям с файловой структурой относятся:

- ☐ навигация по файловой структуре;
- ☐ запуск программ и открытие документов;
- ☐ создание папок;
- ☐ копирование файлов и папок;
- ☐ перемещение файлов и папок;
- ☐ удаление файлов и папок;
- ☐ переименование файлов и папок;
- ☐ создание ярлыков.

Система окон Компьютер

Все операции с файлами и папками в Windows 7 можно выполнять несколькими различными способами. Каждый выбирает себе те приемы, которые ему кажутся наиболее удобными. Обычно с приобретением опыта работы на компьютере совокупность используемых приемов меняется.

Простейшие приемы работы с файловой структурой предоставляет иерархическая система окон папок, берущая свое начало от известной нам папки **Компьютер**. Диски, представленные в окне этой папки, можно открыть, а потом разыскать на них любые нужные папки и файлы. Копирование и перемещение файлов и папок из одной папки в другую можно выполнять путем перетаскивания их значков из окна одной папки в окно другой. Для удаления объектов можно использовать перетаскивание на значок **Корзины**, а можно пользоваться контекстным меню, которое открывается при щелчке правой кнопкой мыши на объекте. Для создания в папке ярлыка документа или программы можно использовать специальное перетаскивание или команду **Создать** ► **Ярлык** из контекстного меню.

При таком подходе к операциям с файловой структурой следует иметь в виду несколько замечаний.

1. В Windows 7 на экране обычно присутствует только одно окно папки. Если в окне папки открыть вложенную папку, то ее окно замещает предыдущее. Это неудобно, если надо выполнять операции перетаскивания между окнами. Чтобы каждая папка открывалась в собственном окне, надо включить следующий переключатель: Пуск ► Панель управления ► Оформление и персонализация ► Параметры папок ► Общие ► Открывать каждую папку в отдельном окне.
2. При перетаскивании значков объектов между папками, принадлежащими одному диску, автоматически выполняется *перемещение* объектов. Если нужно выполнить копирование, используют специальное перетаскивание.
3. При перетаскивании значков объектов между папками, принадлежащими разным дискам, автоматически выполняется *копирование* объектов. Если нужно выполнить перемещение, используют специальное перетаскивание.

Приемы работы с файловой системой

Рассмотрим работу с файловой системой с использованием области переходов папки (Проводника), которую мы рассматривали выше. Для того чтобы открыть окно навигации по файловой системе, можно воспользоваться значком Компьютер, можно сделать щелчок правой кнопкой мыши по кнопке Пуск и выбрать в появившемся контекстном меню команду Открыть проводник. Еще один способ — выполнить команду Пуск ► Все программы ► Стандартные ► Проводник. На рис. 5.3 вы можете видеть структуру папок в области переходов.

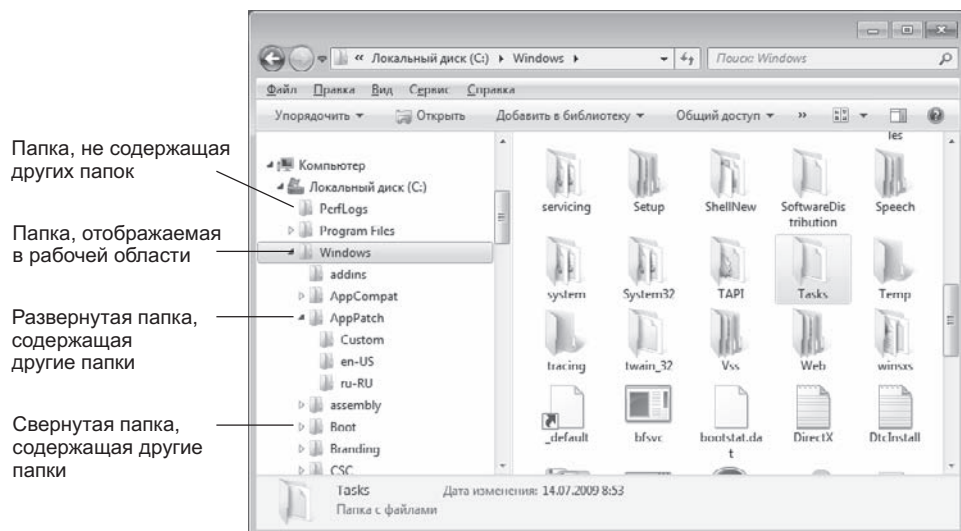


Рис. 5.3. Структура папок в области переходов

Навигация по файловой структуре. Цель навигации состоит в обеспечении доступа к нужной папке и ее содержимому. Мы специально не говорим о том, что

цель навигации — это *поиск* нужных файлов и папок, поскольку для этой операции есть специальные средства.

Навигацию по файловой структуре выполняют с помощью области переходов, на которой показана структура папок. Папки могут быть *развернуты* или *свернуты*, а также *раскрыты* или *закрыты*. Если папка имеет вложенные папки, то на левой панели рядом с папкой отображается *узел*, отмеченный знаком светлого треугольника. Щелчок на узле разворачивает папку, при этом значок узла меняется на темный треугольник. Таким же образом папки и сворачиваются.

Для того чтобы раскрыть папку, надо щелкнуть на ее значке. Содержимое раскрытой папки отображается на правой панели — в рабочей области. Одна из папок на левой панели раскрыта всегда. Закрыть папку щелчком на ее значке невозможно — она закроется автоматически при раскрытии любой другой папки.

Запуск программ и открытие документов. Эта операция выполняется двойным щелчком на значке программы или документа в рабочей области. Если нужный объект на правой панели не показан, надо выполнить навигацию на левой панели и найти папку, в которой он находится.

Создание папок. Чтобы создать новую папку, сначала следует на левой панели раскрыть папку, внутри которой она будет создана. После этого надо перейти на правую панель, щелкнуть правой кнопкой мыши на свободном от значков месте и выбрать в контекстном меню пункт Создать ► Папку. На правой панели появится значок папки с названием Новая папка. Название выделено и в таком состоянии его можно редактировать. После того как папка будет создана, она войдет в состав файловой структуры, отображаемой на левой панели.

Копирование и перемещение файлов и папок. Папку, из которой происходит копирование, называют *источником*. Папку, в которую происходит копирование, называют *приемником*. Копирование выполняют методом перетаскивания значка объекта с правой панели на левую.

Первая задача — найти и раскрыть папку-источник, чтобы на правой панели был виден копируемый объект. Вторая задача — найти на левой панели папку-приемник, но раскрывать ее не надо. Далее объект перетаскивают с правой панели на левую и помещают на значок папки-приемника. Эта операция требует аккуратности, поскольку попасть одним значком точно на другой не всегда просто. Для контроля точности попадания надо следить за названием папки-приемника. В тот момент, когда наведение выполнено правильно, подпись под значком меняет цвет, и кнопку мыши можно отпускать.

Если и папка-источник, и папка-приемник принадлежат одному диску, то при перетаскивании выполняется перемещение, а если разным — то копирование. В тех случаях, когда нужно обратное действие, выполняют специальное перетаскивание при нажатой правой кнопке мыши.

Удаление файлов и папок. Работа начинается с навигации. На левой панели открывают папку, содержащую удаляемый объект, а на правой панели выделяют нужный объект (или группу объектов).

Удаление можно выполнять несколькими способами. Классический способ — с помощью команды Файл ► Удалить из строки меню (если ни один объект не выделен, эта команда не активируется). Более удобный способ — воспользоваться

контекстным меню. Щелкните правой кнопкой мыши на удаляемом объекте и выберите в контекстном меню пункт **Удалить**. Однако самый удобный способ удаления выделенного объекта состоит в использовании клавиши **DELETE** клавиатуры.

Использование манипуляторов, таких как мышь, — это важное достоинство графических операционных систем. Однако профессионалами давно отмечено, что наивысшая производительность труда и минимальное утомление при работе достигаются при максимальном использовании клавиатуры. Для команд, представленных в строке меню, часто приводятся клавиатурные комбинации, которыми эти команды можно выполнить. Обращайте на них внимание, запоминайте их и старайтесь постепенно переходить к их использованию. Это один из приемов закрепления навыков профессиональной работы с компьютером.

Создание ярлыков объектов. Ярлыки объектов можно создавать двумя способами: методом специального перетаскивания (вручную) или с помощью специальной программы-мастера (автоматически). С приемом специального перетаскивания мы уже знакомы. Объект выбирается на правой панели и перетаскивается при нажатой правой кнопке мыши на значок нужной папки на левой панели. В момент отпускания кнопки на экране появляется меню, в котором надо выбрать пункт **Создать ярлык**.

Второй способ (с использованием мастера) менее нагляден, но во многих случаях более удобен. *Мастерами* в системе Windows называют специальные программы, работающие в режиме диалога с пользователем. Диалог строится по принципу «запрос — ответ». Если на все запросы от программы даны корректные ответы, программа автоматически выполнит черновую работу.

1. Для того чтобы запустить Мастер создания ярлыка, надо щелкнуть правой кнопкой мыши в окне той папки, в которой создается ярлык объекта.
2. В открывшемся контекстном меню следует выбрать пункт **Создать ► Ярлык** — произойдет запуск мастера.
3. В диалоговом окне мастера имеется командная строка, в поле которой следует ввести путь доступа к объекту, для которого создается ярлык, например `\Windows\System32\Calc.exe` — путь доступа к стандартной программе Калькулятор. Разумеется, пользователь не может помнить пути доступа ко всем нужным объектам, поэтому ввод адреса автоматизирован. Для этого служит командная кнопка **Обзор**.
4. При щелчке на кнопке **Обзор** открывается диалоговое окно **Обзор файлов и папок**. Это стандартное средство для установления пути доступа к объекту.

Нужную папку и файл разыскивают примерно так же, как на левой панели программы Проводник. Выбирают диск, на котором расположен искомый файл (в нашем случае это диск **C:**), затем разворачивают все вышележащие папки. Список файлов отображается в этом окне ниже имени соответствующей папки.

Разыскав нужный объект, его выделяют и щелкают на кнопке **ОК**. Путь доступа к объекту автоматически заносится в командную строку мастера создания ярлыка.

5. Переход к очередному диалоговому окну мастера выполняют щелчком на командной кнопке **Далее**.
6. В очередном окне мастера вводят название ярлыка, например: **Калькулятор**. Если это последнее окно мастера, то кнопка **Далее** сменяется кнопкой **Готово**. Щелчок на этой кнопке приводит к выполнению заданной операции.

Приемы повышения эффективности в работе с файловой структурой

Приемы, которые здесь описаны, являются общесистемными. Они относятся не только к окнам папок, но и к большинству окон приложений.

Использование буфера обмена для работы с объектами. Система Windows создает и обслуживает на компьютере невидимую для пользователя область памяти, называемую *буфером обмена*. Этой областью можно и нужно уметь пользоваться. В любой момент времени в ней можно хранить только один объект.

Принцип работы с буфером обмена очень прост.

1. Открываем папку-источник. Выделяем щелчком нужный объект.
2. *Копируем* или *забираем* объект в буфер. В первом случае объект остается в папке-источнике и может быть размножен. Во втором случае он удаляется из папки-источника, но может некоторое время храниться в буфере. Последняя операция называется также *вырезанием* объекта.
3. Открываем папку-приемник и помещаем в нее объект из буфера обмена.

Три указанные операции (**Копировать**, **Вырезать** и **Вставить**) можно выполнять разными способами. Классический прием состоит в использовании пункта **Правка** в строке меню, в Windows 7 удобнее пользоваться командами кнопки **Упорядочить** (рис. 5.4).

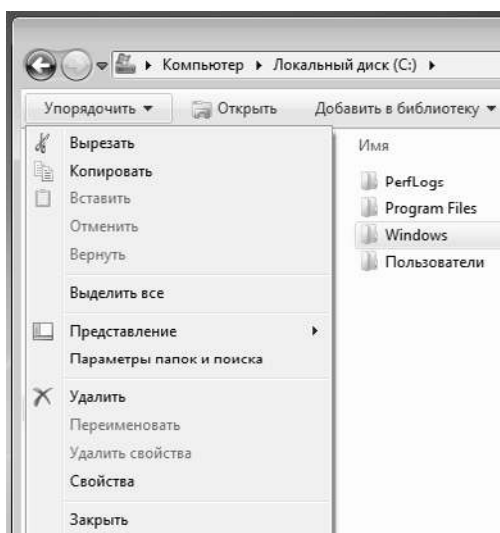


Рис. 5.4. Команды **Вырезать**, **Копировать**, **Вставить**

Самый же эффективный способ работы с буфером обмена состоит в использовании комбинаций клавиш клавиатуры:

- ☐ CTRL+C — копировать в буфер;
- ☐ CTRL+X — вырезать в буфер;
- ☐ CTRL+V — вставить из буфера.

Эти приемы работают во всех приложениях Windows и их стоит запомнить. Через буфер обмена можно переносить фрагменты текстов из одного документа в другой, можно переносить иллюстрации, звукозаписи, видеофрагменты, файлы, папки и вообще любые объекты. Буфер обмена — мощное средство для работы с приложениями и документами в Windows.

В буфере обмена всегда может находиться только один объект. При попытке поместить туда другой объект предыдущий объект перестает существовать. Поэтому буфер обмена не используют для длительного хранения чего-либо. Поместив объект в буфер, немедленно выполняют вставку из буфера в нужное место.

Групповое выделение объектов. Для многих операций (удаление, копирование, перемещение и т. п.) требуется выделить не один объект, а несколько. До сих пор мы использовали для выделения щелчок мыши, но он позволяет выделить только один объект. Для группового выделения при щелчке надо держать нажатой клавишу SHIFT или CTRL.

Если при щелчке держать нажатой клавишу CTRL, то выделение нового объекта не снимает выделение с объектов, выделенных ранее. Так можно выделить любую произвольную группу. Выделение при нажатой клавише CTRL действует как переключатель, то есть повторный щелчок на выделенном объекте снимает выделение.

Если выделяемые объекты расположены подряд, то можно воспользоваться клавишей SHIFT. В этом случае при нажатой клавише щелкают на первом выделяемом объекте группы и на последнем. Все промежуточные объекты выделяются автоматически. Для того чтобы использовать этот прием группового выделения, иногда бывает полезно предварительно упорядочить (отсортировать) объекты, представленные в окне.

Представление объектов. В системе Windows можно управлять тем, как представляются объекты в окнах папок или на правой панели программы Проводник. Существует пять типов представления объектов:

- ☐ Плитка;
- ☐ Значки;
- ☐ Список;
- ☐ Таблица;
- ☐ Содержимое.

Выбор метода представления выполняют либо с помощью команд строки меню (пункт Вид), либо с помощью командной кнопки Изменить представление на панели инструментов. Командная кнопка Изменить представление действует как переключатель, автоматически изменяющий способ представления объектов в окне. Если же надо самостоятельно выбрать способ представления, то рядом с этой кнопкой есть раскрывающаяся кнопка, щелчок на которой раскрывает список возможных режимов (рис. 5.5).

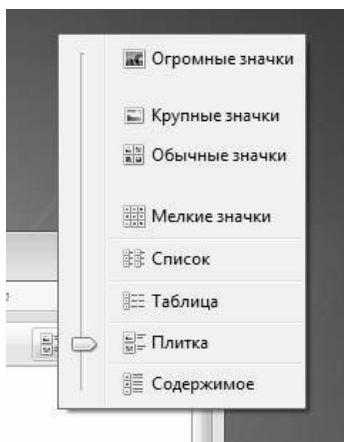


Рис. 5.5. Выбор представления объектов

Режим Плитка применяют в тех случаях, когда в папке находится небольшое количество уникальных объектов (например, программных файлов), каждый из которых важен. В этом режиме отображается не только имя и значок файла, но и некоторые другие его характеристики, зависящие от типа файла.

Режим Значки применяют, когда количество объектов в папке велико и в предыдущем режиме в окне помещается слишком мало значков.

Режим Список применяют в тех случаях, когда в окне присутствуют однотипные объекты, имеющие одинаковые значки. В этом случае содержание объекта характеризует не форма значка, а подпись под ним.

Режим Таблица применяют в тех случаях, когда важны дополнительные свойства объектов, такие как размер, дата создания и т. п. Этот режим интересен также тем, что предоставляет особые возможности по упорядочению объектов в окне.

Режим Содержимое позволяет выводить дополнительную информацию об объектах, подходит для тех случаев, когда количество объектов невелико. Объединяет возможности режимов Таблица и Значки.

Упорядочение объектов. Под упорядочением понимают прежде всего сортировку. В системе Windows 7 существует несколько метода сортировки, наиболее часто используемые: Имя, Тип, Размер и Дата изменения. Метод и направление упорядочения выбирают с помощью команды строки меню Вид ► Сортировать.

Другим способом сортировки объектов является использование заголовочной строки таблицы объектов (при табличном представлении). Столбец, по которому осуществляется сортировка, выделяется значком треугольника в верхней части. Этот же треугольник показывает направление сортировки (по убыванию или по возрастанию). Для смены направления сортировки достаточно щелкнуть по нужному заголовку столбца повторно.

Для сортировки объектов можно воспользоваться соответствующими командами контекстного меню папки.

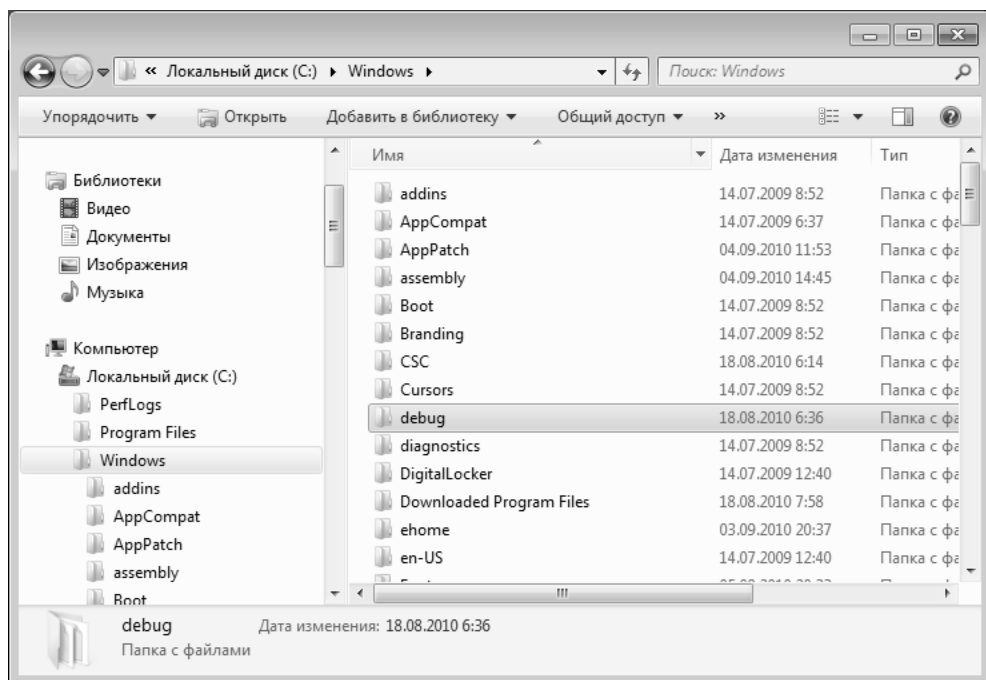


Рис. 5.6. Сортировка таблицы по имени объекта

Если используется метод сортировки **Имя**, объекты в окне располагаются в алфавитном порядке в соответствии с именами связанных с ними файлов (рис. 5.6). Когда при упорядочении во внимание принимается **Тип**, объекты располагаются тоже в алфавитном порядке, но в соответствии с расширениями имен связанных с ними файлов. Вариант **Размер** применяют перед проведением служебных операций. Например, перед очисткой жесткого диска с целью высвобождения рабочего пространства удобно знать, какие объекты наиболее ресурсоемки.

Пункт **Изменен** используют при поиске файлов, изменявшихся в последние дни, или, наоборот, при поиске файлов, не изменявшихся очень долго. Есть вероятность, что документы, не востребованные в течение длительного периода, могут оказаться малонужными и их стоит отправить в архив.

5.4. Использование Главного меню

Структура Главного меню

Главное меню — один из основных системных элементов управления Windows 7. Оно отличается тем, что независимо от того, насколько Рабочий стол перегружен окнами запущенных процессов, доступ к Главному меню удобен всегда — оно открывается щелчком на кнопке **Пуск**. С помощью Главного меню можно запустить все программы, установленные под управлением операционной системы или

зарегистрированные в ней, открыть последние документы, с которыми выполнялась работа, получить доступ ко всем средствам настройки операционной системы, а также доступ к поисковой и справочной системам Windows 7.

Главное меню — необходимый элемент управления для завершения работы с операционной системой. В нем имеется пункт **Завершение работы**, использование которого необходимо для корректного завершения работы с системой перед выключением питания.

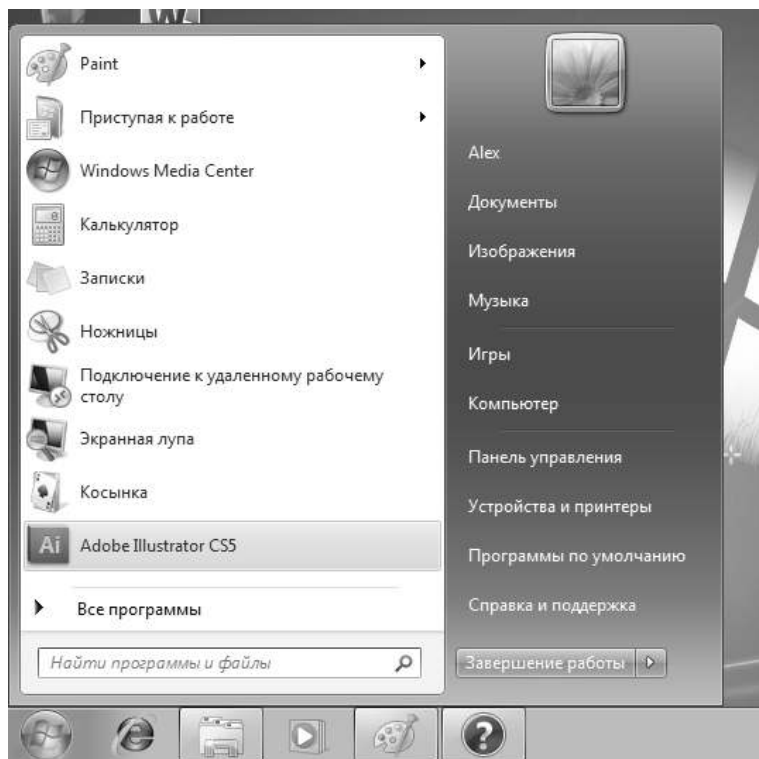


Рис. 5.7. Главное меню

В структуру Главного меню входят три раздела (рис. 5.7).

Левая панель содержит список программ, ее основную область занимают ярлыки программ, которые используются чаще всего, а так же ярлыки, которые пользователь закрепил в Главном меню. Пункт **Все программы** дает доступ к списку установленных программ.

В левой нижней части меню находится строка поиска.

В правой панели находится список наиболее часто используемых объектов системы. В частности, это папки **Документы**, **Компьютер**, ссылки для доступа к **Панели управления**, справочной системе. В нижней части правой панели можно найти команды, позволяющие завершить работу с системой.

В табл. 5.1 вы можете найти описание команд Главного меню.

Таблица 5.1. Структура Главного меню Windows 7

Пункт Главного меню	Назначение	Примечание
Все программы	Открывает доступ к иерархической структуре, содержащей указатели для запуска приложений, установленных на компьютере. Для удобства пользования указатели объединяются в категории. Если категория имеет значок в виде треугольной стрелки, в ней имеются вложенные категории. Раскрытие вложенных категорий выполняется простым зависанием указателя мыши	Указатели, присутствующие в Главном меню, имеют статус ярлыков, а их категории — статус папок. Соответственно, указатели можно копировать и перемещать между категориями, перетаскивать на Рабочий стол и в окна папок. Это один из простейших способов создания ярлыка для недавно установленной программы
Значок с изображением, которое выбрал пользователь при создании учетной записи (верхняя часть правой панели)	Щелкнув мышью по этому значку, вы увидите окно настройки параметров текущего пользователя	
Имя пользователя	В этой строке отображается имя текущего пользователя. Щелчок по ней приводит к открытию персональной папки пользователя	
Документы, Изображения, Музыка	Отрывает доступ к папкам, которые система использует для хранения документов, музыкальных файлов, изображений	Папки, ссылки на которые находятся в Главном меню, система предлагает автоматически в качестве папок по умолчанию при сохранении новых документов. Однако пользователь может самостоятельно определять места, где ему удобно хранить файлы
Игры	Открывает папку, содержащую ярлыки для запуска встроенных игр	Игры, встроенные в Windows, — это не только развлечение. Их можно рассматривать как тренажеры использования мыши для начинающих пользователей.
Компьютер	Открывает окно Компьютер	
Панель управления (Устройства и принтеры, Программы по умолчанию)	Открывает доступ к основным средствам настройки Windows, в частности к логической папке Панель управления. Кнопки, названия которых приведены в скобках, дают быстрый доступ к соответствующим разделам Панели управления	При активной работе с компьютером приходится настолько часто использовать обращение к папке Панель управления, что целесообразно создать для нее ярлык на Рабочем столе

продолжение ⇨

Таблица 5.1 (продолжение)

Пункт Главного меню	Назначение	Примечание
Справка и поддержка	Пункт входа в справочную систему Windows 7	
Завершение работы	Корректное средство завершения работы с операционной системой. При нажатии на кнопку Завершение работы инициирует выключение ПК. При нажатии на кнопку с треугольником в правой части основной кнопки открывает меню, которое, в зависимости от особенностей системы, может содержать следующие команды: ☐ Перезагрузка; ☐ Сон; ☐ Блокировать; ☐ Выйти из системы; ☐ Смена пользователя	Если закрыты все окна процессов, завершить работу с Windows можно комбинацией клавиш ALT+F4. Спящий режим позволяет «заморозить», а затем восстановить состояние компьютера, хотя не все конфигурации оборудования это допускают и не все программы это хорошо переносят
Найти программы и файлы	Набрав в этом поле ключевые слова, по которым хотите найти файлы или папки, вы увидите результаты поиска в верхней части левой панели меню	

5.5. Установка и удаление приложений Windows

В операционной системе Windows 7 есть несколько способов установки приложений, но основным является метод, основанный на использовании установочных файлов.

Перед началом установки нового приложения следует закрыть все работающие программы и все открытые документы. В некоторых случаях необходимо закрывать и ряд фоновых процессов (их наличие может отображаться в виде значков панели индикации на правом краю Панели задач).

Особенности спецификации Windows

Приступая к установке приложений, необходимо знать особенности операционной системы, связанные с *совместным использованием ресурсов*, и помнить, что процедура установки непроверенных программных средств относится к категории потенциально опасных.

Принцип совместного использования ресурсов лежит в основе спецификации Windows, и в области программного обеспечения он приводит к тому, что разные приложения могут использовать общие программные ресурсы. Так, например, в большинстве приложений Windows можно встретить одинаковые элементы оформления и управления (окна, кнопки, раскрывающиеся списки, меню, флажки,

переключатели и многое другое). Одинаковы и приемы управления ими, и методы их использования. С точки зрения приложений это означает, что их многие компоненты обрабатываются одним и тем же программным кодом. Поэтому в Windows принято выделять стереотипные программные фрагменты и группировать их в динамические библиотеки, к которым открыт доступ для разных программ (динамические библиотеки имеют расширение имени файла .DLL).

При установке новых приложений вместе с ними устанавливаются только те программные ресурсы, которые нужны для работы данного приложения, но отсутствуют на данном компьютере (то есть не зарегистрированы в его операционной системе). Поэтому для установки новых приложений очень важно, чтобы они проходили правильную регистрацию. В состав дистрибутивных комплектов большинства современных приложений входят специальные устанавливающие программы (**Setup.exe**, **Install.exe**). Обычно они корректно выполняют все необходимые операции по регистрации в системе новых приложений и компонентов.

Стандартные средства установки приложений

Обычно установочный файл автоматически запускается при установке в дисковод оптических дисков CD или DVD с программой. Также установочный файл может быть загружен из Интернета и запущен пользователем самостоятельно.

Сравнительно новый способ установки программ называется веб-установкой. При его использовании на компьютер нужно загрузить небольшую программу, которая называется **веб-установщик**. После запуска программа предложит пользователю выбрать компоненты основного программного продукта. По завершении выбора нужные компоненты будут загружены из Интернета, с сервера поставщика ПО и установлены.

После установки приложения нередко требуется перезагрузить компьютер. В системе Windows 7 необходимость перезагрузки возникает реже, чем в предыдущих версиях операционных систем семейства Windows, но тоже может потребоваться. Это одна из причин, по которой до начала установки закрывают все открытые приложения и документы.

Необходимость перезагрузки связана с особенностью операционной системы. Некоторые операции выполняются удобно и безопасно только в момент завершения работы системы или на начальном этапе ее загрузки, когда большинство модулей системы еще не активированы.

Удаление приложений Windows

Удаление ранее установленных приложений Windows производится средствами раздела Панели управления, к которому можно получить доступ по следующему пути: **Пуск ► Панель управления ► Удаление программы**.

Открыв его, следует выбрать из списка запись о программе, которую требуется удалить, выделить ее и нажать на кнопку **Удалить/Изменить**. После этого будет запущено автоматическое средство удаления программы.

Некоторые программы помещают ярлык для запуска процесса удаления в свою группу в меню **Пуск**. Найдя в папке программы (**Пуск ► Все программы ►**

Наименование_программы) нужный ярлык, можно открыть его и добиться того же эффекта, как при использовании Панели управления.

Удаление редко бывает полным. Скорее всего, какие-то компоненты останутся. Чаще всего остаются некоторые папки (как правило, пустые). Компоненты, не удаленные автоматически, следует удалить вручную. Рекомендуется удалять их в Корзину и наблюдать за компьютером в течение нескольких дней. Если после этого работоспособность прочих программ не нарушается, эти компоненты можно удалить и из Корзины.

5.6. Установка оборудования

В общем случае оборудование подключается к компьютеру дважды: аппаратно и программно. Под *аппаратным подключением* понимают физическое соединение с компьютером либо с помощью гнезд на материнской плате, либо с помощью внешних разъемов стандартных портов на задней стенке системного блока. Бывает и смешанное подключение, когда *интерфейсная плата* нового устройства вставляется в слот материнской платы и при этом создается новый (нестандартный) порт, разъем которого выходит на заднюю стенку. Таким способом подключают, как правило, устройства, требующие высокой скорости передачи данных, например сканеры или сетевые устройства.

Под *программным подключением* понимают установку программы-драйвера, являющейся посредником между операционной системой и устройством. При установке драйвера происходит выделение операционной системой части ресурсов новому устройству, а также регистрация устройства и его драйвера в реестре операционной системы.

Однако в общем правиле есть и исключения. Такие «стандартные» устройства, как жесткие диски, дисководы гибких дисков и клавиатура, не требуют драйверов, поскольку сведения о том, как с ними работать, уже имеются в базовой системе ввода-вывода (BIOS). Они должны распознаваться и работать еще до загрузки операционной системы. То же относится и к монитору, и к видеоадаптеру, но без драйверов они распознаются только как простейшие стандартные модели. Для того чтобы использовать все функциональные возможности конкретной модели, драйвер установить необходимо.

Несколько менее «стандартными» устройствами считаются мышь и дисковод CD-ROM. Они не всегда распознаются средствами BIOS, но после загрузки операционной системы Windows 7 уже считаются стандартными устройствами и обслуживаются драйверами, имеющимися в ее составе; однако если речь идет о необычных моделях, особый драйвер для них может потребоваться.

Абсолютное большинство прочих устройств требуют наличия программного драйвера. При продаже аппаратного обеспечения общепринято прикладывать к устройству программные драйверы на компакт-диске. В отсутствие такой возможности можно воспользоваться библиотекой драйверов, входящей в состав операционной системы. Если библиотека не поддерживает конкретную модель устройства, необходимый драйвер можно получить в Интернете на сервере фирмы, изготовившей оборудование, или на сервере компании Microsoft, где имеется

коллекция драйверов устройств для операционных систем, выпускаемых этой компанией. Даже для старых и надежно работающих устройств рекомендуется периодически (два раза в год) посещать сервер изготовителя и получать обновленную версию драйвера. Своевременное обновление драйверов устройств повышает эффективность работы оборудования, улучшает совместимость с программным обеспечением и повышает общую надежность системы.

Средства программной установки оборудования

В подавляющем большинстве случаев Windows 7 автоматически распознает подключенные к ней устройства и устанавливает драйверы для них, при необходимости загружая их из Интернета.

Если с устройством, которое вы хотите установить, поставляется компакт-диск с драйверами, его можно поместить в дисковод и следовать указаниям программы автоматической установки.

Базовое программное средство установки оборудования запускается из Панели инструментов: Пуск ► Панель инструментов ► Добавление устройства. С его помощью можно установить большую часть оборудования, хотя в общем правиле есть исключения.

Для установки и обновления драйверов устройств можно воспользоваться окном Диспетчер устройств (Пуск ► Панель управления ► Оборудование и звук ► Диспетчер устройств).

Порядок установки оборудования

Как правило, оборудование, которое вы будете подключать к компьютеру, снабжается кратким руководством пользователя, которое содержит описание порядка подключения устройства к компьютеру и порядок установки программного обеспечения для него.

Подавляющее большинство современных внешних устройств подключается к компьютеру по интерфейсу USB. Этот интерфейс поддерживает «горячее» подключение, то есть устройство можно подсоединить к работающему ПК. Однако, для того чтобы установка прошла правильно, учитывайте требования поставщика устройства, изложенные в руководстве пользователя.

Например, в руководстве по установке принтера может быть сказано, что сначала нужно установить программное обеспечение (содержащее, в том числе, драйверы принтера) с прилагающихся к устройству дисков и только после этого подключить устройство к системе. Выполняйте рекомендации из руководств пользователя и вы сможете правильно установить любое устройство.

Нередко нестандартные устройства до установки специализированных драйверов могут работать в Windows как обычные устройств, без использования дополнительных функций. И только установка специализированного ПО позволит воспользоваться всеми возможностями этих устройств. Например, это касается нестандартных мышей. Такая мышь может содержать дополнительные кнопки, колесики и без установки дополнительного ПО будет работать как обычная мышь.

Если устройство не было опознано при запуске, надо воспользоваться Мастером установки оборудования. Мастер запускается командой Пуск ► Панель управления ► Добавление устройства. Следуйте его указаниям, при необходимости пользуясь справочной системой Windows.

По окончании процесса установки оборудования компьютер следует перезагрузить и выполнить проверку на наличие конфликтов. Для проверки наличия конфликтов и опознанных, но не установленных устройств, а также для получения подробной информации об установленных в системе устройствах воспользуйтесь Диспетчером устройств.

Практическое занятие

Упражнение 5.1. Отработка приемов управления с помощью мыши

1. **Зависание.** Слева на Панели задач имеется кнопка Пуск. Это элемент управления Windows, называемый *командной кнопкой*. Наведите на нее указатель мыши и задержите на некоторое время — появится *всплывающая подсказка*: Пуск.

Справа на Панели задач расположена *панель индикации*. На этой панели, в частности, расположен индикатор *системных часов*. Наведите на него указатель мыши и задержите на некоторое время — появится всплывающая подсказка с показаниями *системного календаря*.

2. **Щелчок.** Наведите указатель мыши на кнопку Пуск и щелкните левой кнопкой — над ней откроется Главное меню Windows. Меню — это один из элементов управления, представляющий собой список возможных команд. Команды, представленные в меню, выполняются щелчком на соответствующем пункте. Все команды, связанные с элементами управления, выполняются одним обычным щелчком.

Однако у щелчка есть и другое назначение. Его применяют также для *выделения* объектов. Разыщите на Рабочем столе значок Компьютер и щелкните на нем. Значок и подпись под ним изменят цвет. Это произошло выделение объекта. Объекты выделяют, чтобы подготовить их к дальнейшим операциям.

Щелкните на другом объекте, например на значке Корзина. Выделение значка Компьютер снимется, а вместо него выделится значок Корзина. Если нужно снять выделение со всех объектов, для этого достаточно щелкнуть на свободном от объектов месте Рабочего стола.

3. **Двойной щелчок.** Двойной щелчок применяют для *использования* объектов. Например, двойной щелчок на значке, связанном с приложением, приводит к запуску этого приложения, а двойной щелчок на значке документа приводит к открытию данного документа в том приложении, в котором он был создан. При этом происходит одновременно и запуск этого приложения. Относительно документа оно считается *родительским*.

В системе Windows 7 с одним и тем же объектом можно выполнить много разных действий. Например, файл с музыкальной записью можно воспроизвести (причем в разных приложениях), его можно отредактировать, можно скопировать на другой носитель или удалить. Сколько бы действий ни было

возможно с объектом, всегда существует одно *основное действие*. Оно и выполняется двойным щелчком.

Выполните двойной щелчок на значке **Компьютер**, и на экране откроется одноименное окно **Компьютер**, в котором можно увидеть значки дисков, подключенных к компьютеру, значок **Панели управления** и другие значки.

Если нужно закрыть окно, надо щелкнуть один раз на *закрывающей кнопке*, которая находится в правом верхнем углу окна. Закрывающая кнопка — это элемент управления, и для работы с ним достаточно одного щелчка.

4. **Щелчок правой кнопкой.** Щелкните правой кнопкой на значке **Мой компьютер**, и рядом с ним откроется элемент управления, который называется *контекстным меню*. У каждого объекта Windows свое контекстное меню. Состав его пунктов зависит от свойств объекта, на котором произошел щелчок. Для примера сравните содержание контекстного меню объектов **Мой компьютер** и **Корзина**, обращая внимание на их различия (рис. 5.8–5.9).

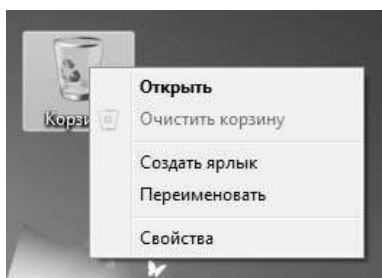


Рис. 5.8. Контекстное меню Корзины

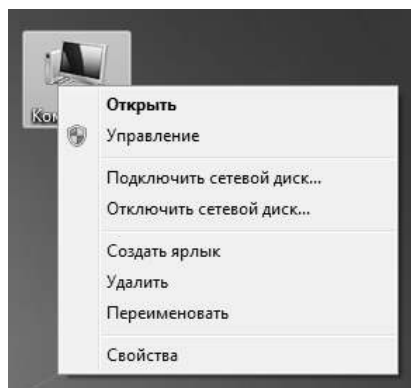


Рис. 5.9. Контекстное меню Компьютера

Доступ к контекстному меню — основное назначение щелчка правой кнопкой. В работе с объектами Windows (особенно с новыми) щелчок правой кнопкой используется очень часто.

Контекстное меню чрезвычайно важно для работы с объектами операционной системы. Выше мы говорили, что двойной щелчок позволяет выполнить только то действие над объектом, которое считается *основным*. В противоположность этому в контекстном меню приведены *все действия*, которые можно выполнить над данным объектом. Более того, во всех контекстных меню любых объектов имеется пункт **Свойства**. Он позволяет просматривать и изменять свойства объектов, то есть выполнять настройки программ, устройств и самой операционной системы.

5. **Перетаскивание.** Перетаскивание — очень мощный прием для работы с объектами операционной системы. Наведите указатель мыши на значок **Компьютер**. Нажмите левую кнопку и, не отпуская ее, переместите указатель — значок **Компьютер** переместится по поверхности Рабочего стола вместе с ним.

Откройте окно **Компьютер**. Окно можно перетаскивать с одного места на другое, если «подцепить» его указателем мыши за строку заголовка. Так прием перетаскивания используют для оформления рабочей среды.

6. **Протягивание.** Откройте окно **Компьютер**. Наведите указатель мыши на одну из рамок окна и дождитесь, когда он изменит форму, превратившись в двунаправленную стрелку. После этого нажмите левую кнопку и переместите мышь. Окно изменит размер. Если навести указатель мыши на правый нижний угол окна и выполнить протягивание, то произойдет изменение размера сразу по двум координатам (по вертикали и горизонтали).

Изменение формы объектов Windows — полезное, но не единственное использование протягивания. Нередко этот прием используют для *группового выделения* объектов. Наведите указатель мыши на поверхность Рабочего стола, нажмите кнопку мыши и протяните мышь вправо-вниз — за указателем потянется прямоугольный контур выделения. Все объекты, которые окажутся внутри этого контура, будут выделены одновременно.

7. **Специальное перетаскивание.** Наведите указатель мыши на значок **Компьютер**, нажмите правую кнопку мыши и, не отпуская ее, переместите мышь. Этот прием отличается от обычного перетаскивания только используемой кнопкой, но дает иной результат. При отпускании кнопки не происходит перемещение объекта, а вместо этого открывается так называемое *меню специального перетаскивания*. Содержимое этого меню зависит от перемещаемого объекта. Для большинства объектов в нем четыре пункта (**Копировать**, **Переместить**, **Создать ярлыки** и **Отменить**). Для таких *уникальных* объектов, как **Мой компьютер** или **Корзина**, в этом меню только два пункта: **Создать ярлыки** и **Отменить**.

Мы убедились, что, несмотря на то что стандартная мышь имеет только две кнопки, с их помощью можно реализовать весьма разнообразные приемы управления. Мы узнали наиболее характерные особенности этих приемов и их общепринятое назначение. В основе идеологии Windows лежит принцип, согласно которому базовые приемы управления операционной системой должны использоваться и при управлении ее приложениями. Знание общесистемных приемов пригодится в работе с любыми приложениями данной системы.

Упражнение 5.2. Изучение приемов работы с объектами

1. Откройте папку **\Документы** (**Пуск** ▶ **Документы**).
2. Щелчком на раскрывающей кнопке разверните окно на полный экран.
3. В строке меню дайте команду **Файл** ▶ **Создать** ▶ **Папку**. Убедитесь в том, что в рабочей области окна появился значок папки с присоединенной надписью **Новая папка**.
4. Щелкните правой кнопкой мыши на свободной от значков рабочей области окна текущей папки. В открывшемся контекстном меню выберите команду **Создать** ▶ **Папку**. Убедитесь в том, что в пределах окна появился значок папки с надписью **Новая папка (2)**.

5. Щелкните правой кнопкой мыши на значке Новая папка. В открывшемся контекстном меню выберите пункт Переименовать. Дайте папке содержательное имя, например Экспериментальная. Аналогично переименуйте папку Новая папка (2). Убедитесь в том, что операционная система не допускает существования в одной папке (\Документы) двух объектов с одинаковыми именами. Дайте второй папке имя Мои эксперименты.
6. Восстановите окно папки \Документы до нормального размера щелчком на восстанавливающей кнопке.
7. Откройте окно Компьютер. В нем откройте окно с содержимым жесткого диска (C:). Пользуясь полосами прокрутки, разыщите в нем папку \Windows и откройте ее двойным щелчком. Ознакомьтесь с текстом предупреждающего сообщения о том, что изменение содержания этой системной папки может быть потенциально опасным. Включите отображение содержимого папки щелчком на ссылке Отображать содержимое этой папки. В открывшемся содержимом разыщите значок папки \Temp и откройте ее (эта папка считается папкой временного хранения данных, и экспериментировать с ее содержимым можно без опасений). Перетаскиванием переместите папку \Экспериментальная из папки \Документы в папку C:\Windows\Temp. Специальным перетаскиванием переместите папку \Мои эксперименты в папку C:\Windows\Temp и по окончании перетаскивания выберите пункт Переместить в открывшемся контекстном меню.
8. Откройте окно C:\Windows\Temp. Щелчком выделите значок папки \Экспериментальная. При нажатой клавише CTRL щелчком выделите значок папки \Мои эксперименты. Убедитесь в том, что в рабочей области одновременно выделено два объекта (групповое выделение).
9. Заберите выделенные объекты в буфер обмена комбинацией клавиш CTRL+X. Убедитесь в том, что их значки стали полупрозрачными в рабочей области папки.
10. Откройте окно папки \Документы. Вставьте в него объекты, находящиеся в буфере обмена (CTRL+V).
11. Выделите значки папок \Экспериментальная и \Мои эксперименты в папке \Документы. Щелкните правой кнопкой мыши и в открывшемся контекстном меню выберите пункт Удалить. В открывшемся диалоговом окне подтвердите необходимость удаления объектов. Закройте окно папки \Мои документы.
12. Двойным щелчком на значке откройте окно Корзина. Убедитесь, что в нем находятся значки удаленных папок \Экспериментальная и \Мои эксперименты. Выделите оба значка. Щелкните правой кнопкой мыши и в открывшемся контекстном меню выберите пункт Восстановить. Закройте Корзину.
13. Откройте окно папки \Документы. Убедитесь в том, что в нем восстановились значки папок \Экспериментальная и \Мои эксперименты. Выделите оба значка. Удалите их с помощью клавиши DELETE при нажатой клавише SHIFT. В открывшемся диалоговом окне подтвердите необходимость удаления объектов. Закройте окно папки \Документы.

14. Откройте окно Корзины. Убедитесь в том, что объекты, удаленные при нажатой клавише SHIFT, не поступили в Корзину. Закройте Корзину.

Мы научились создавать новые папки с помощью строки меню и контекстного меню, научились давать папкам осмысленные имена, познакомились с тремя приемами копирования и перемещения объектов между окнами папок (перетаскиванием, специальным перетаскиванием и с использованием буфера обмена). Мы освоили приемы группового выделения объектов, удаления объектов в Корзину и окончательного удаления, минуя Корзину.

Упражнение 5.3. Работа с файловой структурой в программе Проводник (на панели навигации)

1. Включите персональный компьютер, дождитесь окончания загрузки операционной системы.
2. Запустите программу Проводник с помощью Главного меню (Пуск ► Все программы ► Проводник). Обратите внимание на то, какая папка открыта на левой панели Проводника в момент запуска. Это должна быть папка \Библиотеки.
3. Перейдите в папку Документы, сделав двойной щелчок на ее значке в рабочей области папки Библиотеки. На правой панели создайте новую папку \Экспериментальная.
4. На левой панели разверните папку \Документы одним щелчком на значке узла (светлом треугольнике). Обратите внимание на то, что *раскрытие* и *разворачивание* папок на левой панели — это разные операции. Убедитесь в том, что на левой панели в папке \Документы образовалась вложенная папка \Экспериментальная.
5. Откройте папку \Экспериментальная на левой панели. На правой панели не должно отображаться никакое содержимое, поскольку эта папка пуста.
6. Создайте на правой панели Проводника новую папку \Мои эксперименты внутри папки \Экспериментальная. На левой панели убедитесь в том, что рядом со значком папки \Экспериментальная образовался узел, свидетельствующий о том, что папка имеет вложенные папки. Разверните узел и рассмотрите образовавшуюся структуру на левой панели.
7. На левой панели разыщите папку \Windows и разверните ее.
8. На левой панели внутри папки \Windows разыщите папку для временного хранения объектов — \Temp, но не раскрывайте ее.
9. Методом перетаскивания переместите папку \Экспериментальная с правой панели на левую — в папку C:\Windows\Temp. Эту операцию надо выполнять аккуратно. Чтобы «попадание» было точным, следите за цветом надписи папки-приемника. При точном наведении надпись меняет цвет — в этот момент можно отпускать кнопку мыши при перетаскивании. Еще труднее правильно «попасть в приемник» при перетаскивании групп выделенных объектов. Метод контроля тот же — по выделению надписи.
10. На левой панели откройте папку C:\Windows\Temp. На правой панели убедитесь в наличии в ней папки \Экспериментальная.

11. Разыщите на левой панели Корзину и перетащите папку \Экспериментальная на ее значок. Раскройте Корзину и проверьте наличие в ней только что удаленной папки. Закройте окно программы Проводник.

Мы научились выполнять навигацию с помощью левой панели программы Проводник и изучили приемы копирования и перемещения объектов методом перетаскивания между панелями. Те, кто считает, что с левой панелью Проводника работать не очень удобно, могут исполнять все операции, пользуясь только правой панелью. При этом используют следующие свойства Проводника:

- возможность копирования и перемещения объектов через буфер обмена;
- программу Проводник можно запустить несколько раз — соответственно, на Рабочем столе можно иметь несколько правых панелей, между которыми удобно выполняются все операции обмена.

Исследовательская работа

Задание. Исследование методов запуска программы Проводник

В операционной системе Windows 7 большинство операций можно выполнить многими разными способами. На примере программы Проводник мы исследуем различные приемы запуска программ.

1. Щелкните правой кнопкой мыши на кнопке Пуск и в открывшемся контекстном меню используйте пункт Открыть проводник. Обратите внимание на то, какая папка открыта на левой панели в момент запуска.
2. Щелкните правой кнопкой мыши на значке Компьютер и в открывшемся контекстном меню используйте пункт Открыть. Обратите внимание на то, какая папка открыта на левой панели в момент запуска.
3. Проверьте контекстные меню всех значков, открытых на Рабочем столе. Установите, для каких объектов контекстное меню имеет средства запуска Проводника, и выясните, какая папка открывается на левой панели в момент запуска.
4. Выполните запуск Проводника через пункт Все программы Главного меню.
5. Выполните запуск Проводника через пункт Пуск ► Все программы ► Стандартные ► Выполнить Главного меню.
6. Выполните запуск Проводника через ярлык папки \Документы (Пуск ► Документы ► *щелчок правой кнопкой мыши* ► Открыть).
7. Выполните запуск Проводника с Рабочего стола (предварительно на Рабочем столе следует создать ярлык Проводника).
8. Выполните запуск Проводника с Панели быстрого запуска.
9. Заполните отчетную таблицу по образцу.

Метод запуска Проводника	Используемый элемент управления	Папка открытия
Через контекстное меню кнопки Пуск	Кнопка Пуск	C:\Windows\Главное меню

Настройка операционной системы Windows

Панель управления > Все элементы панели управления > Поиск в панели управления

Настройка параметров компьютера Просмотр: Крупные значки

Windows CardSpace	Автозапуск	Администрирование
Архивация и восстановление	Брандмауэр Windows	Восстановление
Гаджеты рабочего стола	Дата и время	Датчик расположения и другие датчики
Диспетчер устройств	Диспетчер учетных данных	Домашняя группа
Защитник Windows	Звук	Значки области уведомлений
Клавиатура	Мышь	Панель задач и меню "Пуск"
Параметры индексирования	Параметры папок	Персонализация
Подключения к удаленным рабочим	Почта	Приступая к работе
Программы и компоненты	Программы по умолчанию	Распознавание речи

Внешний вид стартового окна Панели управления можно настраивать, выбирая один из способов просмотра его элементов с помощью списка Просмотр на панели инструментов окна. На рис. 6.1 мы выбрали способ просмотра Крупные значки. По умолчанию способ просмотра установлен в значение Категория.

При представлении содержимого Панели управления по категориям ее элементы разбиты на логические группы, наиболее часто используемые элементы выделены. При представлении в виде значков деления на группы нет. Используя этот вид представления можно быстро найти нужный объект в том случае, если вы точно знаете его название и предназначение или не знаете точно, в какой группе он расположен при представлении элементов Панели инструментов по категориям.

Если вы затрудняетесь в поиске необходимого объекта Панели управления, вы можете воспользоваться стандартным полем поиска, которое расположено в верхнем правом углу окна Панели.

6.1. Настройка средств ввода-вывода данных

Настройка клавиатуры

Настройку клавиатуры выполняют в диалоговом окне Свойства: Клавиатура, которое открывают двойным щелчком на значке Клавиатура в окне Панель управления. На вкладке Скорость представлены средства настройки параметров функции *автотоповтора символов* (величина задержки перед началом повтора символов и темп повтора), а также средства управления частотой мерцания курсора.

Настройка мыши

Настройку мыши выполняют в диалоговом окне Свойства: Мышь, которое открывают с помощью значка Мышь в окне Панель управления. На вкладке Кнопки мыши представлены средства назначения левой или правой кнопки функций *основной кнопки*, а также средства настройки интервала времени между щелчками, при котором два отдельных щелчка интерпретируются как один двойной.

На вкладке Указатели представлены средства для выбора схемы указателей мыши. Схема указателей представляет собой именованную совокупность настроек формы указателей мыши, сохраняемую в отдельном файле.

На вкладке Параметры указателя представлены средства для управления *чувствительностью* мыши. Чувствительность мыши определяется величиной экранного перемещения указателя при заданном перемещении прибора. Выбор чувствительности зависит от типа мыши или другого манипулятора, а также от привычного режима работы конкретного пользователя (от характерного размаха движений мыши в процессе управления).

На этой же вкладке имеются средства управления видимостью указателя. Есть возможность скрывать указатель во время работы с клавиатурой, а также задействовать средства подсветки указателя при работе с малоконтрастными дисплеями, например некоторыми жидкокристаллическими дисплеями портативных компьютеров.

На вкладке Колесико можно настраивать параметры колесика мыши. В частности, здесь можно установить параметры вертикальной и горизонтальной прокрутки (для мышей, поддерживающих наклон колесика).

Настройка стиля управления операционной системой

Начиная с Windows 98, операционные системы семейства Windows поддерживают несколько *стилей управления*. До сих пор мы рассматривали только так называемый *классический стиль управления*, восходящий к принципам Windows 95. Он характерен тем, что объекты выделяют одним щелчком, а открывают двумя щелчками.

Другой стиль управления характерен для работы в Интернете. Он подразумевает, что объекты выделяют простым наведением указателя, а открывают одним щелчком. Данный стиль позволяет несколько повысить производительность в ряде операций с объектами, но не очень удобен при проведении групповых операций.

Выбор того или иного стиля управления выполняют включением переключателя на вкладке Общие диалогового окна Свойства папки (Пуск ► Панель управления ► Параметры папок).

Для использования классического стиля надо установить переключатель **Открывать двойным**, а выделять одним щелчком. Для использования стиля, характерного для Интернета, установите переключатель **Открывать одним щелчком**, выделять указателем. В этом случае подписи значков выделяются подчеркиванием. Это может происходить всегда или, для более привычного вида Рабочего стола и окон, только при наведении указателя. Соответствующая настройка также выбирается установкой переключателя.

6.2. Настройка оформления рабочей среды

Персонализация настроек Windows

Большинство настроек рабочей среды выполняется с помощью специального средства Персонализация. Для доступа к нему используют либо соответствующий компонент Панели управления (Пуск ► Панель управления ► Персонализация), либо одноименный пункт в контекстном меню Рабочего стола. В обоих случаях открывается диалоговое окно Персонализация, представленное на рис. 6.2. Его средства позволяют:

- ☐ избрать фоновый рисунок Рабочего стола;
- ☐ настроить оформление окон Windows и их элементов;
- ☐ подключить и настроить заставку экрана;
- ☐ задать схему звукового оформления и схему указателей мыши;
- ☐ настроить параметры графического дисплея;
- ☐ сохранить пакет настроек в качестве темы оформления Windows.

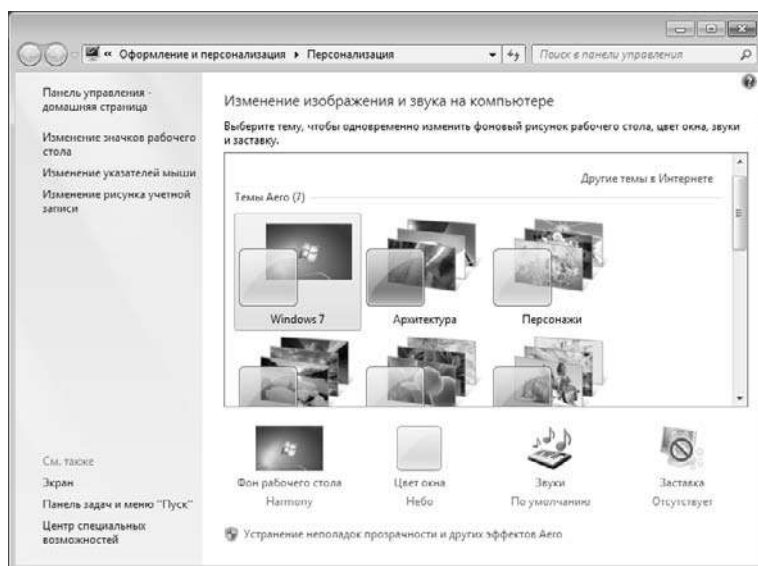


Рис. 6.2. Компонент Персонализация — основное средство настройки рабочей среды Windows

Настройка фона Рабочего стола

Операционная система Windows позволяет в качестве фона Рабочего стола заливку цветом заданного тона или ранее подготовленную и сохраненную иллюстрацию. Иллюстрация должна иметь формат, принятый в Интернете. Подключение выполняется с помощью компонента **Фоновый рисунок рабочего стола** (рис. 6.3), который, в свою очередь, запускается из окна компонента **Персонализация**.

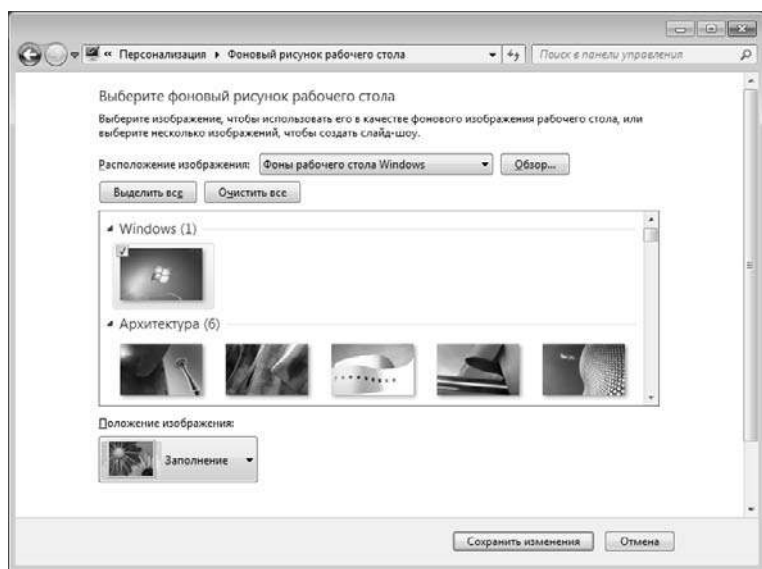


Рис. 6.3. Средство выбора и настройки фона Рабочего стола

В левом верхнем углу миниатюры изображения, которое в данный момент является фоновым рисунком Рабочего стола, установлен флажок. Если установить такие флажки у нескольких изображений, они будут периодически сменять друг друга на Рабочем столе.

По умолчанию рисунки, используемые в качестве фоновых, хранятся в системной папке `C:\Windows\Web\Wallpaper\`. Достаточно разместить здесь файл с изображением, и оно будет доступно для выбора в качестве фонового. Выбор выполняют с помощью раскрывающегося списка **Расположение рисунка**, где следует выбрать пункт **Фоны рабочего стола Windows**, как показано на рис. 6.3.

При подключении фонового рисунка предоставляется возможность выбора способа его размещения: **Заполнение**, **По размеру**, **Растянуть**, **Замостить**, **По центру**. В первом случае рисунок заполняет экран, увеличиваясь или уменьшаясь до размеров экрана. Причем искажений рисунка не происходит, но, например, если соотношение сторон рисунка не соответствует соотношению сторон монитора, на Рабочем столе будет видна лишь центральная часть рисунка. Во втором случае рисунок увеличивается или уменьшается до тех пор, пока одна из его сторон не окажется равной одной из сторон экрана. Вариант **Растянуть** масштабирует рисунок, растягивая (или сжимая) его до размеров экрана. Вариант **Замостить** вызывает многократное повторение копий избранного рисунка по всему полю Рабочего

стола. Размещение По центру исключает искажение рисунка в результате масштабирования, но при этом допускает появление черных полей по краям.

Если один компьютер находится в совместном использовании несколькими сотрудниками, целесообразно располагать файлы фоновых рисунков не в специализированной папке, а в личных папках пользователей: C:\Users\{Имя_пользователя}\Pictures\. Иллюстрации, хранящиеся здесь, подключаются в качестве фоновых выбором пункта Библиотека изображений в раскрывающемся списке Расположение рисунка.

При желании для хранения фоновых рисунков можно использовать также произвольную папку локального компьютера. Выбор папки выполняют с помощью командной кнопки Обзор.

В заключение укажем на техническую возможность использования в качестве фона однотонной цветовой заливки. Она реализуется выбором пункта Сплошные цвета в раскрывающемся списке Расположение рисунка.

Настройка экранной заставки

Экранные заставки — это динамические изображения, воспроизведение которых включается автоматически при отсутствии в течение заданного времени событий, вызванных пользователем. Первоначальное назначение заставок состояло в том, чтобы снизить угрозу «выгорания люминофора» на тех участках экрана, которые подвержены особо длительному стационарному воздействию электронного луча. Результатом этого эффекта было образование в местах длительного воздействия луча бурых пятен. Современным мониторах эффект «выгорания люминофора» не грозит, но экранные заставки продолжают использовать как средство сокрытия экранной информации от посторонних наблюдателей в период отсутствия владельца компьютера на рабочем месте.

Для настройки экранной заставки служит диалоговое окно Параметры экранной заставки. Его можно открыть из окна компонента Персонализация. Представленные здесь средства позволяют выбрать вид заставки (раскрывающийся список Заставка), задать период, по истечении которого происходит ее автоматический запуск (счетчик Интервал), а также настроить параметры ее воспроизведения (командная кнопка Параметры).

Если сеанс работы текущего пользователя защищен паролем, то для отключения заставки его необходимо предъявить. При отсутствии защиты заставка отключается активацией мыши или клавиатуры.

Тематическое оформление рабочей среды

Совокупность визуальных и звуковых настроек интерфейса Windows рассматривается как тема Рабочего стола. Тема включает в себя набор реквизитных значков Рабочего стола, параметры настройки шрифтового и цветового оформления, комплект указателей мыши, а также экранную заставку и схему звуковых оповещений.

По умолчанию вместе с операционной системой на компьютер устанавливаются два набора тем оформления: базовые (упрощенные) и темы Aero (расширенные).

Среди базовых тем можно отметить следующие:

- ❑ Классическая: ее стиль оформления соответствует оформлению устаревшей системы Windows 98. Эта тема рекомендуется к применению для малопроизводительных компьютерных систем.
- ❑ Контрастные темы. Эти темы рекомендуется применять пользователям, имеющим проблемы со зрением.
- ❑ Windows 7 — упрощенный стиль. Упрощенная тема, которая рекомендуется к применению на компьютерах со слабым аппаратным обеспечением.

Расширенные темы можно использовать тогда, когда нет каких-либо предпосылок к применению базовых тем.

При изменении любого элемента действующей темы операционная система рассматривает возникшую совокупность настроек как новую тему оформления. Новая тема выводится в верхней части окна списка тем, при необходимости новую тему можно сохранить, воспользовавшись соответствующими элементами управления (рис. 6.4).

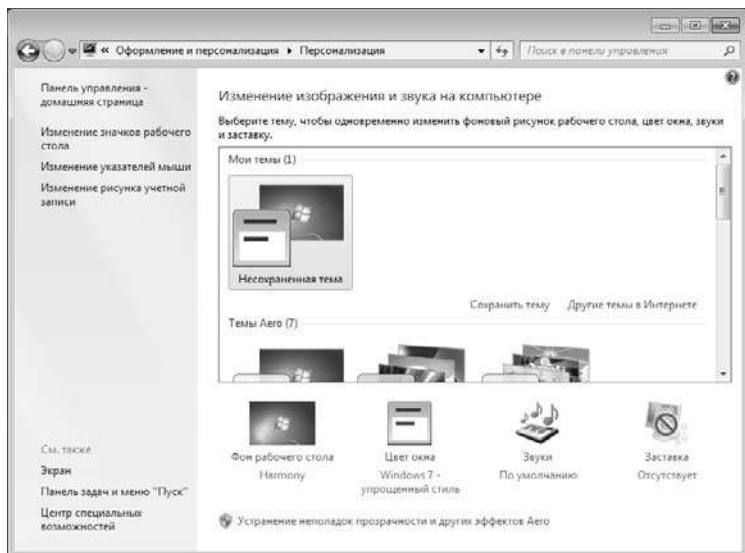


Рис. 6.4. Несохраненная тема в списке тем

Оформление окон Windows

За цветовое и шрифтовое оформление окон Windows отвечают средства диалогового окна Цвет и внешний вид окна (Пуск ► Панель управления ► Персонализация ► Цвет окна).

Окно Цвет и внешний вид окна служит для тонкой настройки отдельных элементов оформления окон Windows. Порядок настройки такой: сначала в раскрывающемся списке Элемент выбирается элемент оформления, подлежащий настройке, после чего для него задаются размер, цвет, шрифт, а также параметры шрифтового начертания.

Для некоторых элементов оформления допускается многоцветное оформление с помощью градиентных растяжек (плавных переходов) между двумя заданными краевыми цветами. Выбор цвета осуществляют в раскрывающейся палитре с фиксированным количеством цветов. Любой цвет палитры можно определить самостоятельно — доступ к цветовой матрице открывает командная кнопка Другой.

Настройка реквизитных значков Рабочего стола

Некоторые значки Рабочего стола выполняют функции стандартных реквизитов. В частности, к ним относятся значки: Компьютер, Сеть, Корзина и др. Составом и внешним видом реквизитных значков можно управлять.

В списке задач, в левой части окна Персонализация (Пуск ► Настройка ► Персонализация ► *имеется задача* ► Изменение значков рабочего стола. Ее ссылка открывает диалоговое окно Параметры значков рабочего стола. Требуемый пакет значков формируют установкой соответствующих флажков, а чтобы сменить внешний вид значка, следует этот значок выделить и выбрать новый образ из палитры с помощью кнопки Сменить значок. Возврат к стандартному представлению значка выполняется кнопкой Обычный значок.

Регулировка размера подписей под значками

Регулировка подписей под значками может требоваться для дисплеев с высокой плотностью пикселей на единицу длины, а также при наличии у пользователя ограничений по зрению. Существует два приема, позволяющих увеличить размер символов в подписях: через список задач компонента Персонализация или с помощью средств дополнительного оформления графических элементов Windows.

Первый прием опирается на аппаратные средства компьютера и выполняется настройкой драйвера видеоадаптера. Это «жесткое» решение. Им целесообразно пользоваться для систем, имеющих жесткие, стабильные конфигурации.

1. Запустите компонент Персонализация (Пуск ► Панель управления ► Оформление и персонализация ► Изменение размеров текста и других элементов).
2. Включите переключатель Средний 125%

Второй прием позволяет гибко управлять размером подписей под значками. Он опирается только на программные средства операционной системы. Это «гибкое» решение. Им целесообразно пользоваться на компьютерах, конфигурация которых способна изменяться часто и непредсказуемо.

3. Откройте диалоговое окно Цвет и внешний вид окна (Пуск ► Панель управления ► Оформление и персонализация ► Персонализация ► Цвет окна).
4. В раскрывающемся списке Элемент выберите настраиваемый элемент — Значок.
5. Увеличьте размер шрифта и утвердите изменение кнопкой ОК.

Размер символов традиционно измеряется в полиграфических пунктах. По умолчанию в операционной системе принята величина 9 пт. Увеличение этой величины до 10 пт обычно бывает достаточно, но если это не так, задайте другую величину.

Настройка параметров дисплея

К настраиваемым параметрам дисплея относятся:

- ☐ величина экранного разрешения — измеряется количеством точек полноэкранного изображения по горизонтали и вертикали;
- ☐ качество цветопередачи — выражается количеством одновременно отображаемых цветов. Может также называться глубиной цвета или цветовым разрешением и выражаться количеством двоичных разрядов, принятых для кодирования цвета отдельной точки;
- ☐ дополнительные параметры, такие как настройка **Clear Type (технологии, улучшающей изображение на ЖК-дисплеях)**, частота экранной развертки.

Оптимальные значения обоих параметров зависят от свойств видеоадаптера и монитора. В частности, для дисплеев, основанных на ЖК-мониторах, эти параметры должны совпадать с физическими характеристиками прибора. Контроль и настройка параметров выполняется с помощью средства настройки экрана (Пуск ► Панель управления ► Оформление и персонализация ► Экран).

Настройка звуковых схем Windows

Звуковое оформление Windows основано на том, что системным событиям, как аппаратным, так и программным, а также событиям приложений и событиям, связанным с действиями пользователя, можно сопоставить определенные звуковые клипы. Теоретически, озвучить можно любое событие, но обычно так поступают только с важнейшими событиями, имеющими оперативное значение. Например, по умолчанию к озвучиваемым событиям относятся: открытие и закрытие окон Windows, удаление объектов в Корзину, поступления сообщений электронной почты.

Звуковое сопровождение событий принято настраивать в пакетном режиме — с помощью звуковых схем. Звуковая схема — это именованная совокупность настроек, определяющая соответствие между заданными событиями и сопоставленными им звукозаписями.

Для настройки звукового оформления рабочей среды предназначен компонент Звук (Пуск ► Панель управления ► Оборудование и звук ► Звук). Элементы управления вкладки Звуки данного диалогового окна (рис. 6.5) позволяют загружать имеющиеся звуковые схемы, редактировать их и сохранять. Несколько стандартных звуковых схем поставляются совместно с операционной системой. Их редактирование осуществляется путем правки назначения звуков системным событиям. Результаты редактирования можно сохранить в виде новой звуковой схемы.

Назначение звуков системным событиям выполняют в списке Программные события (см. рис. 6.5). События, которым уже сопоставлен звуковой клип, отмечены значком громкоговорителя. Данный клип можно воспроизвести кнопкой Проверить. Если необходимо удалить звуковое оформление события, следует выделить его и в раскрывающемся списке Звуки выбрать пункт (Нет).

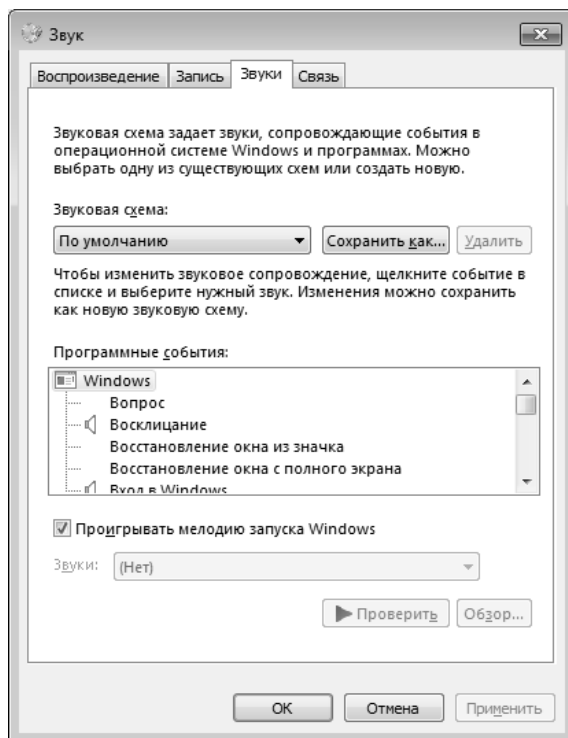


Рис. 6.5. Системное средство контроля, формирования и обслуживания звуковых схем

6.3. Настройка элементов управления Windows

Настройка Панели задач

Панель задач в операционной системе Windows — настраиваемая: ее свойствами можно управлять. В исходном состоянии она примыкает к нижней кромке экрана, но методом перетаскивания ее положение можно изменить. Соответственно, вместе с ней меняют свое положение кнопка Пуск и область уведомлений.

Размер Панели задач можно регулировать протягиванием мыши. Для этого следует привести указатель на внешнюю рамку и дождаться, когда он сменит форму. Предельный размер Панели задач — половина экрана.

Изменение свойств Панели задач осуществляется через пункт Свойства контекстного меню. Чтобы открыть его, следует щелкнуть правой кнопкой мыши где-либо на свободном от кнопок месте. В свою очередь, команда Свойства открывает диалоговое окно Свойства панели задач и меню «Пуск», представленное на рис. 6.6.

Данное окно имеет три вкладки. Настройка Панели задач производится на вкладке Панель задач. Здесь наиболее важны установки двух флажков: Автоматически скрывать панель задач и Закрепить панель задач. Чтобы открыть Панель задач, когда она скрыта, достаточно подвести указатель мыши к тому краю экрана, за которым она находится.

Панель задач операционных систем семейства Windows обладает рядом интересных особенностей. В частности, в ее рамках можно создать ряд вспомогательных инструментальных панелей:

- ☐ Панель адресов Интернета;
- ☐ Панель ссылок на веб-страницы Интернета;
- ☐ Панель объектов Рабочего стола;
- ☐ Панель быстрого запуска;
- ☐ Языковая панель (служит для переключения языков ввода).

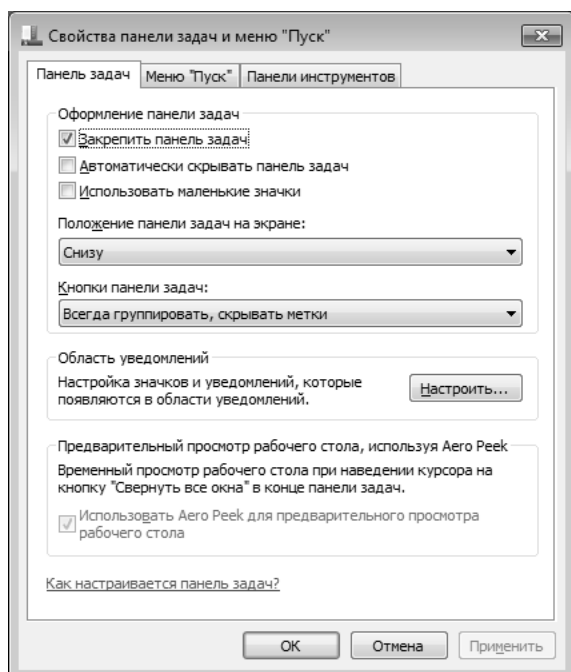


Рис. 6.6. Средство настройки свойств Панели задач и Главного меню Windows

Для создания и удаления этих панелей служит команда **Панели**, имеющаяся в контекстном меню **Панели задач**. Особенно широко используется **Панель быстрого запуска**. На ней обычно располагают ярлыки часто используемых программ.

По завершении настройки **Панели задач**, ее состояние следует закрепить. Для этого служит уже упомянутый флажок **Закрепить панель задач**. Он располагается в контекстном меню **Панели задач** или в диалоговом окне ее свойств. При установке данного флажка дальнейшее изменение настроек **Панели задач** блокируется.

Настройка Главного меню

Главное меню — основной элемент управления в операционных системах Windows. С его помощью можно запустить программу, установленную на компьютере с веро-
ятом операционной системы, открыть документы, с которыми выполнялась работа

в последние дни, а также выполнить большинство системных настроек. Главное меню открывается щелчком на кнопке Пуск (рис. 6.7).

Главное меню имеет многоуровневую структуру. Так, например, при наведении указателя мыши на пункт Все программы открывается система вложенных меню, отображающая распределение программ по категориям. По своим свойствам каждая категория меню Все программы имеет статус папки, а каждый пункт — статус ярлыка. Таким образом, структурой меню программ можно управлять, управляя структурой папок, представляющих его. Простейший способ открыть структуру программного меню для редактирования — воспользоваться пунктом Открыть контекстного меню, вызванного для пункта Все программы. Это касается лишь состава и структуры списка установленных программ. Для того чтобы настраивать другие свойства главного меню, нужно воспользоваться вкладкой Главное меню окна Свойства панели задач и меню «Пуск», которое можно вызвать из контекстного меню Панели задач.

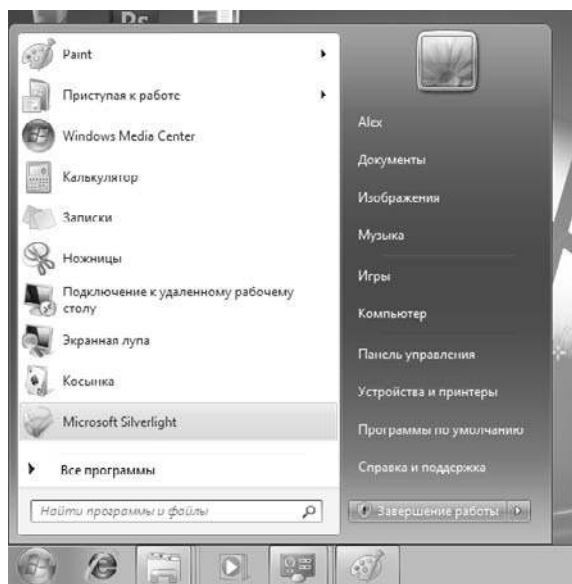


Рис. 6.7. Главное меню — основной элемент управления операционной системы Windows

Настройка свойств Корзины

Корзина представляет собой специальную папку Windows, в которой временно хранятся удаленные объекты. Физически, на жестком диске она представлена скрытой папкой \Recycled, причем для каждого жесткого диска, имеющегося в вычислительной системе, папка \Recycled — своя. Однако логически Корзина представляет собой одну-единственную логическую папку, соответствующую всем физическим папкам \Recycled, имеющимся в компьютерной системе.

Настройку свойств Корзины выполняют в диалоговом окне Свойства: Корзина, которое открывается через контекстное меню Корзины.

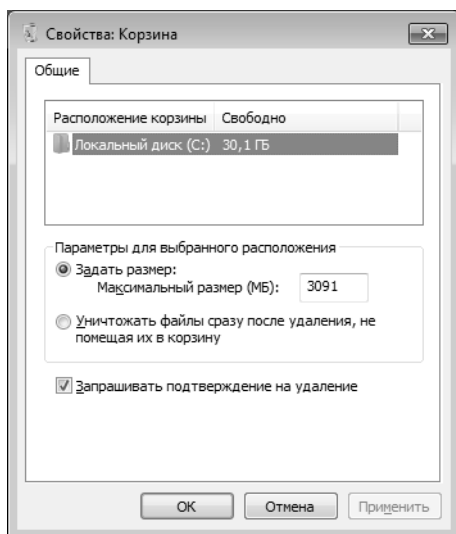


Рис. 6.8. Настройка свойств Корзины Windows

Основным параметром Корзины является ее предельная емкость. Когда объем файлов в Корзине начинает превосходить установленное значение, операционная система автоматически чистит Корзину, окончательно уничтожая файлы, которые были помещены туда раньше всего. Этот параметр выставляется движком и измеряется в процентах от емкости соответствующих дисков (по умолчанию — 10%). Прочие элементы управления диалогового окна свойств Корзины предусматривают возможность удаления объектов без помещения их в Корзину (используется при глобальной расчистке жесткого диска) и возможность отключения сообщения, предупреждающего об удалении объектов.

Настройка свойств окон папок

К основным настраиваемым свойствам окон папок относятся: режим отображения скрытых и системных объектов, а также механизм обзора вложенных папок. Выполняют настройку этих свойств в диалоговом окне Параметры папок. Из Главного меню оно открывается командой Пуск ► Панель управления ► Оформление и персонализация ► Параметры папок. Его можно открыть также из окна любой папки: Сервис ► Параметры папок.

Заметим, что пользователи операционной системы Windows 7 обычно не видят в окне папки строку меню. Чтобы открыть эту строку и получить доступ к команде Сервис, надо нажать клавишу F10 или Alt.

Существует два механизма обзора вложенных папок: либо все вложенные папки открываются в одном и том же окне, либо каждой вложенной папке создается отдельное окно. Первый способ не перегружает Рабочий стол открытыми окнами, а второй упрощает навигацию в структуре папок. Механизм обзора выбирают на вкладке **Общие** диалогового окна **Параметры папок** с помощью переключателей: **Открывать папки в одном и том же окне** или **Открывать каждую папку в отдельном окне** (рис. 6.9).

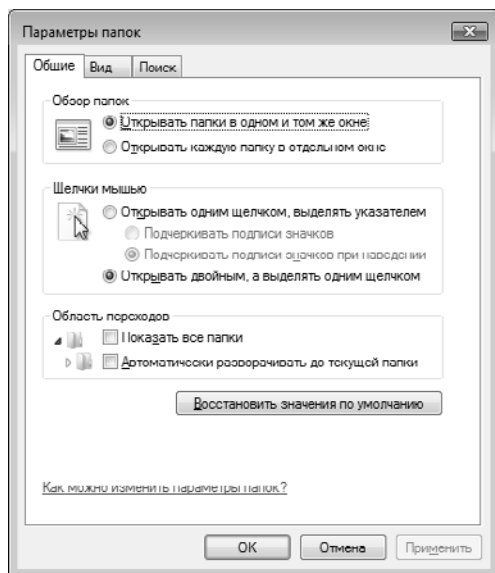


Рис. 6.9. Настройка свойств окон папок

Значки системных и скрытых объектов в окнах папок обычно не отображают: они не требуются при работе с документами и приложениями. При этом о наличии в окне папки скрытых объектов можно судить по записи в строке состояния. Включить отображение значков системных и скрытых объектов рекомендуется перед удалением папок, а также при обслуживании операционной системы. Элементы управления для включения и отключения отображения скрытых и системных объектов находятся на вкладке **Вид** диалогового окна **Параметры папок** в категории **Файлы и папки** ► **Скрытые файлы и папки**.

6.4. Настройка средств автоматизации Windows

Автоматический запуск приложений

Для автоматического запуска приложений после загрузки операционной системы в Windows 7 предусмотрено очень простое средство — специальная папка **Автозагрузка** (**\Главное меню\Все программы\Автозагрузка**). Настройка автоматического запуска приложений выполняется копированием ярлыков запускаемых приложе-

ний в эту папку. Соответственно, отключение автоматического запуска приложения выполняют удалением его ярлыка из папки \Автозагрузка.

С помощью папки \Автозагрузка можно не только запускать приложения, но и открывать документы. Соответственно, в этом случае в папку необходимо предварительно поместить ярлык документа. Открытие документа происходит с одновременным запуском *родительского* приложения, которое предназначено для работы с документами данного типа.

Настройка свойств типов файлов

В основе ряда автоматических операций с файлами лежит тот факт, что операционной системе известно, какое именно приложение должно использоваться для работы с документами того или иного типа. То есть между приложениями Windows и документами Windows существует определенная настраиваемая связь. Она действует через расширение имени файла.

В современных версиях Windows, начиная с операционной системы Windows Vista, связь между приложениями и документами настраивается редактированием свойств значка, соответствующего типу файла. Для выбора приложения, которое требуется использовать для открытия файлов, нужно в окне свойств значка перейти на вкладку Общие и нажать на кнопку Изменить. В результате появится окно Выбор программы, с помощью которого можно настроить связь между приложениями и документами, (рис. 6.10).

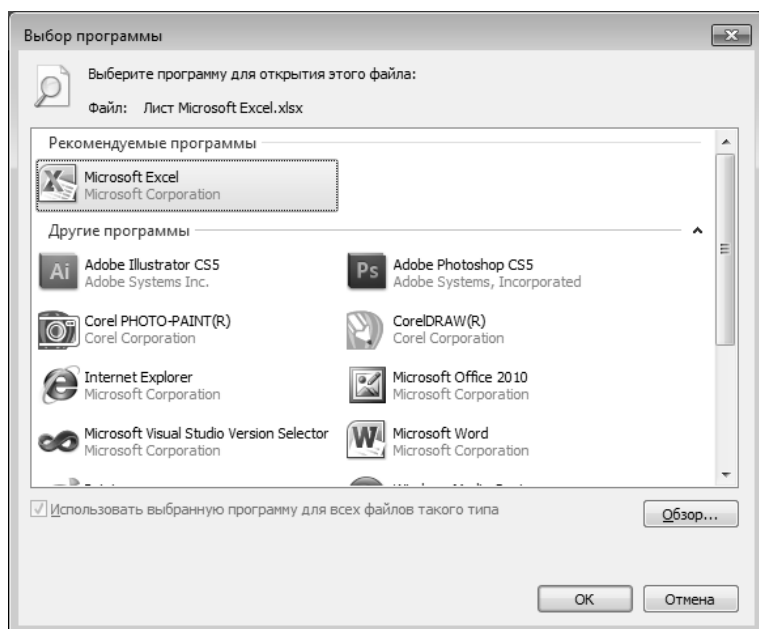


Рис. 6.10. В операционной системе Windows 7 связь между документом и приложением настраивается редактированием свойств значка документа

Если необходимо открыть документ в специфической программе для исполнения какого-то специального действия, следует пользоваться командой **Открыть с помощью** в контекстном меню документа.

Настройка команды **Отправить**

Команда **Отправить** — мощное средство повышения производительности труда. Она имеется в контекстном меню большинства объектов, и с ней связан список контейнеров, которые могут служить адресатами при пересылке текущего объекта. Использование команды **Отправить** — простейший способ для копирования документа на заданный носитель, для отправки его по заданному адресу электронной почты, для создания ярлыка на Рабочем столе и т. п. Пункты меню команды **Отправить** — настраиваемые и редактируемые. Неиспользуемые пункты можно удалить, а вместо них создать другие, более удобные.

Настройка команды **Отправить** выполняется наполнением специальной папки \SendTo ярлыками папок, устройств и каналов связи. Содержимое этой папки специфично для каждого пользователя. Каждый ярлык, присутствующий в данной папке, соответствует одному из пунктов меню команды **Отправить**.

Папка SendTo в Windows 7 находится по следующему пути:

C:\Users\Имя_Пользователя\AppData\Roaming\Microsoft\Windows\SendTo

Для того чтобы быстро открыть эту папку, можно воспользоваться следующей методикой. В меню **Пуск**, в строке поиска, нужно набрать следующую команду `shell:sendto`. В результате выполнения этой команды на экране откроется папка SendTo для текущего пользователя.

Автоматизация очистки жесткого диска

Необходимость в автоматической очистке жесткого диска особенно важна для компьютеров, работающих в круглосуточном режиме, без непосредственного участия пользователей. Если при отсутствии пользователя в работе компьютера возникает исключительная ситуация, связанная с переполнением жесткого диска, система сама должна выполнить необходимые операции очистки. Для этого в нее входит агентское приложение **Очистка диска**.

Агент очистки запускается командой **Пуск ► Все программы ► Стандартные ► Служебные ► Очистка диска**. После запуска программы следует указать имя диска, для которого выполняется настройка. Агент включается в работу автоматически, если операционная система обнаруживает на диске недостаток свободного места.

Состав папок, подлежащих автоматической очистке, задается на вкладке **Очистка диска** (рис. 6.11). Разумеется, далеко не все папки жесткого диска подлежат очистке в автоматическом режиме. Теоретически, папок, в которых не должны храниться невосполнимые данные, не так уж много: они указаны в списке **Удалить следующие файлы**. Заметим, что для разных дисков этот список различен. Он зависит от размещения служебных каталогов.

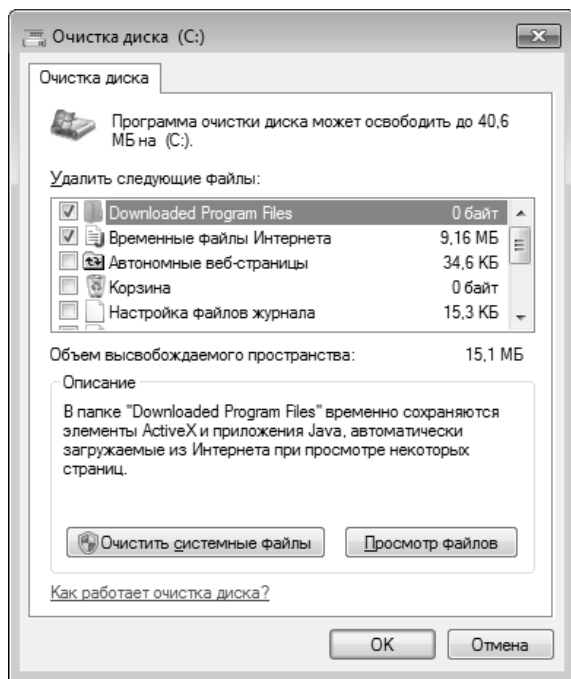


Рис. 6.11. Настройка параметров автоматической очистки диска

Выбор нужных осуществляют установкой соответствующих флажков.

- ☐ **Downloaded Program Files** — папка, в которой хранятся активные объекты, содержащие программный код, принятый из Интернета. Их хранение служит для ускорения загрузки веб-страниц при повторном посещении.
- ☐ **Временные файлы Интернета** — это папка, в которой кэшируются данные, принятые из Интернета при работе со службой WWW. Кэширование служит только для ускорения загрузки веб-страниц при их повторном посещении. Никакой документальной ценности данные, хранящиеся в этой папке, не представляют.
- ☐ **Корзина** — основной кандидат для автоматической очистки.
- ☐ **Временные файлы** — имеется в виду папка \Temp, в которой не принято хранить ценные данные. Приложения нередко автоматически создают в ней свои служебные временные файлы, но не всегда могут их удалить (например, в случае аварийного завершения работы), в результате чего эта папка часто перегружается ненужными отходами.
- ☐ **Эскизы** — уменьшенные изображения рисунков, видеоматериалов, документов, использующиеся в окнах папок для представления содержимого. Удаление эскизов не имеет критических последствий, но вызовет на некоторое время замедление в открытии окон папок, пока система не восстановит эскизы содержимого папок.

Запуск приложений по расписанию

Одним из методов автоматизации работ, выполняемых на компьютере под управлением операционной системы Windows, является запуск приложений по назначенному расписанию.

В Windows 7 функции запуска приложений по расписанию возложены на компонент Планировщик заданий (Пуск ► Панель управления ► Система и безопасность ► Расписание выполнения задач). Основная особенность Планировщика заданий заключается в том, что он позволяет реализовывать очень сложную логику заданий и обслуживает как пользовательские, так и системные задачи.

В основе работы Планировщика заданий лежат четыре понятия:

- ☐ триггеры;
- ☐ действия;
- ☐ условия;
- ☐ параметры.

Триггер определяет момент начала исполнения задания. Различают временные триггеры и событийные. Временной триггер определяет время запуска задания. Соответственно событийный триггер определяет событие, при наступлении (или регистрации в журнале) которого задание начинает исполняться.

Действием, в свою очередь, является операция, которая должна быть выполнена при срабатывании триггера. Такая операция может составлять все задание или только его часть. Планировщик заданий допускает создание сложных заданий. В состав сложного задания может входить до 32 действий.

Условия — это дополнительные ограничения для триггеров. Например, если триггер установлен на запуск задания в определенное время, условие может препятствовать срабатыванию триггера, если компьютер в это время питается от батарей.

Параметры — это дополнительные настройки, уточняющие режим исполнения задания.

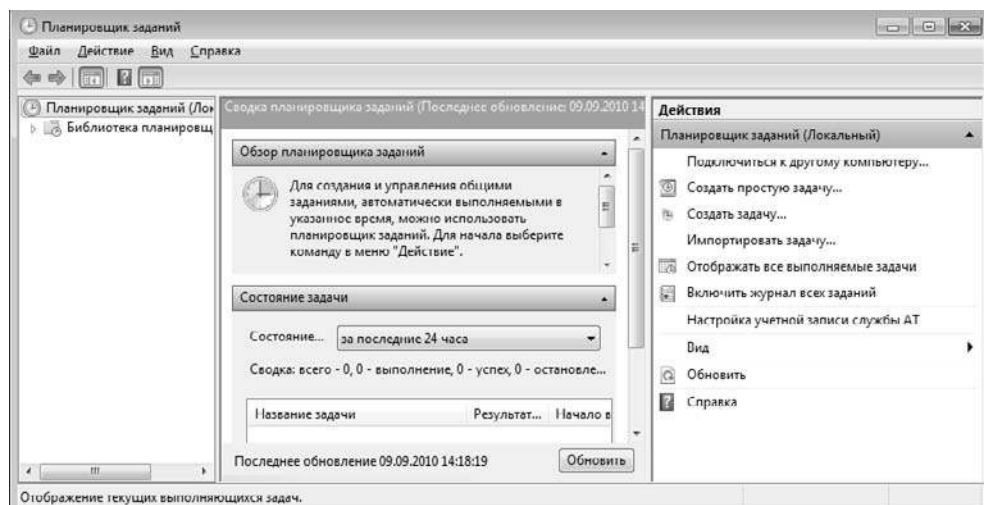


Рис. 6.12. Планировщик заданий

Существует два типа заданий: простые и сложные. Простые задания предназначены для реализации простейшей логики. Они содержат один триггер и одно действие, например, простым является следующее задание: ежедневно в 07:00 в качестве будильника запускать воспроизведение определенной звукозаписи. Создаются простые задания с помощью Мастера, который запускается командой **Создать простую задачу** (рис. 6.12).

В свою очередь, сложные задания позволяют реализовывать более сложную логику и могут содержать несколько действий. Такие задания создают либо вручную (команда **Создать задачу**), либо путем редактирования и дополнительной настройки ранее созданного простого задания.

6.5. Настройка средств поиска данных

Механизмы автоматического поиска данных

Существует два подхода к организации поиска данных на персональном компьютере: традиционный (файловый) и современный (контекстный). Файловый поиск предполагает, что отбор документов выполняется на основе формальных сведений об атрибутах файлов. К ним относятся: имя файла, его тип, дата создания или последнего изменения, размер и другие свойства. Как правило, механизм файлового поиска имеет своей целью обеспечение доступа к известным материалам, для которых неизвестно или забыто место хранения. Чем больше пользователю известно о файле, в котором должен находиться разыскиваемый документ, тем оперативнее и результативнее будет поиск.

Контекстный поиск предполагает, что отбор материалов ведется не по формальным признакам, а по содержанию документов. Более того, контекстный поиск, в отличие от файлового, может проводится не только в файлах, но и в базах данных, в архивах и других структурах данных. Соответственно, целью контекстного поиска является не только восстановление доступа к забытым или утраченным данным, но также анализ содержания документов и приобретение новых знаний. Работа средств контекстного поиска на локальной компьютерной системе во многом похожа на работу поисковых сервисов в Интернете.

Контекстный поиск данных

Механизм контекстного поиска является более передовым достижением по сравнению с механизмом файлового поиска. В семействе ОС Windows он появился, начиная с системы Windows Vista. Контекстный поиск выполняется по ключевым словам. При этом анализируются не только имена и свойства файлов, но также и содержание документов, и содержание записей баз данных.

Контекстный поиск требует значительных вычислительных ресурсов, поэтому он реализуется в несколько этапов. На этапе предварительной подготовки операционная система просматривает содержимое носителей и индексирует найденные данные; эти операции выполняются в фоновом режиме. В свою очередь, отбор документов, соответствующих поисковому заданию, происходит в режиме реального времени. На этом этапе просматриваются не сами документы, а указатели, сформированные в ходе индексации.

Доступ к механизму контекстного поиска осуществляется через Главное меню. Поисковое задание формируется набором ключевых слов, записанных в поле ввода, которое находится в нижней части Главного меню и содержит исходную фразу Найти программы и файлы. Удалять эту фразу нет необходимости — просто начните ввод ключевых слов. С каждым введенным знаком в Главном меню формируются результаты поиска (рис. 6.13).

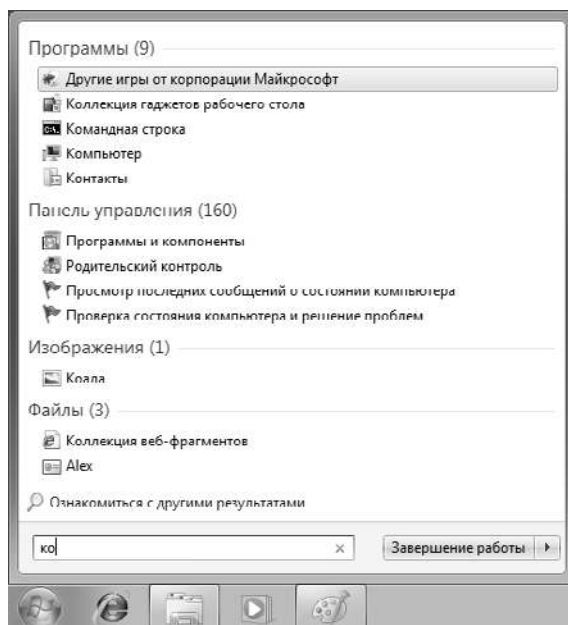


Рис. 6.13. Контекстный поиск документов в индексированных областях

Найденные объекты автоматически распределяются по категориям. На рис. 6.13. можно видеть объекты, которые относятся к категориям Программы, Панель управления, Изображения, Файлы.

При проведении контекстного поиска разрешается использовать подстановочные символы «*» и «?». Они позволяют указать, какие знаки результирующего текста следует считать произвольными (не конкретными).

Символ «*» заменяет произвольное число любых символов (возможно, ни одного). Так, например, по ключевому слову мол* будут отобраны документы, содержащие слова молочный, молоко, молния, молитва и т. д.

В свою очередь, символ «?» заменяет ровно один произвольный символ, и по ключевому слову мол??? могут быть отобраны документы со словами молоко и молния, но не моль и не молот.

Сложный контекстный поиск

Контекстный поиск бывает простым, сложным и расширенным. Простой контекстный поиск мы уже рассмотрели. Поисковое задание для него формируется в виде простого списка ключевых слов.

Сложный контекстный поиск отличается от простого тем, что в качестве поискового задания используется логическое выражение, составленное из ключевых слов и операторов логических отношений AND, NOT, OR. Кроме того, при записи выражения разрешается использовать кавычки и круглые скобки.

Таблица 6.1. Логические отношения в контекстном поиске

Логическое отношение	Действие	Пример
AND	Отбираются документы, содержащие все слова выражения	Авиация AND палубная
NOT	Ключевое слово, указанное непосредственно за словом NOT, не должно присутствовать в документе	Авиация NOT палубная
OR	Результат должен содержать любое из заданных слов, но не обязательно их все	Авиация OR палубная
Кавычки	Ищется точная фраза, заключенная в кавычки	"Авиация палубная"
Скобки	Должны быть найдены все ключевые слова в произвольном порядке и не обязательно рядом друг с другом	(авиация палубная)

Расширенный поиск

Расширенный поиск используют, когда помимо ключевых слов, характеризующих содержание документа, имеются дополнительные сведения об информационном объекте (например, о файле), в котором искомый документ хранится. Средство расширенного поиска доступно через поисковое поле, которое можно найти в папках.

При проведении расширенного поиска можно указать следующие параметры (рис. 6.14).

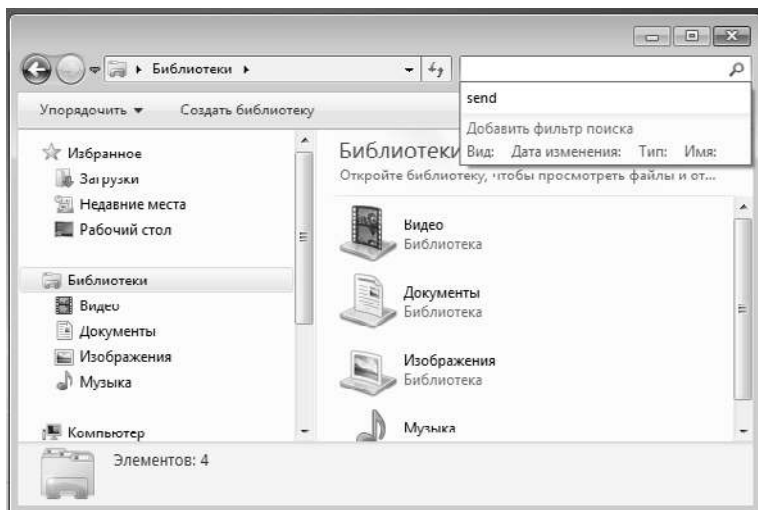


Рис. 6.14. Расширенный поиск

При проведении расширенного поиска помимо ключевых слов, характеризующих содержание документа, возможно указать различные параметры. Их набор зависит от папки, в которой производится поиск. Среди этих параметров можно отметить следующие:

- ☐ вид информационного объекта (документ, сообщение, папка, контакт);
- ☐ диапазон дат создания или последнего изменения;
- ☐ ограничения на размер (имеет смысл для файлов).

Как нетрудно убедиться, технология расширенного поиска позволяет реализовать механизм не только контекстного, но и файлового поиска.

Особенности поиска в папках

В штатном режиме контекстный поиск обеспечивается только в областях, прошедших предварительную индексацию. Это связано с особой трудоемкостью контекстного поиска в неиндексированных областях. Тем не менее, если речь идет о сравнительно небольшой области, например о конкретной папке, возможен контекстный поиск без предварительной индексации.

Поле поиска, о котором мы уже говорили, имеется во всех окнах папок. Реакция системы на ввод ключевых слов в это поле примерно та же, что и при поиске через Главное меню, но поиск выполняется не по всему компьютеру, а лишь в текущей папке и в ее дочерних папках.

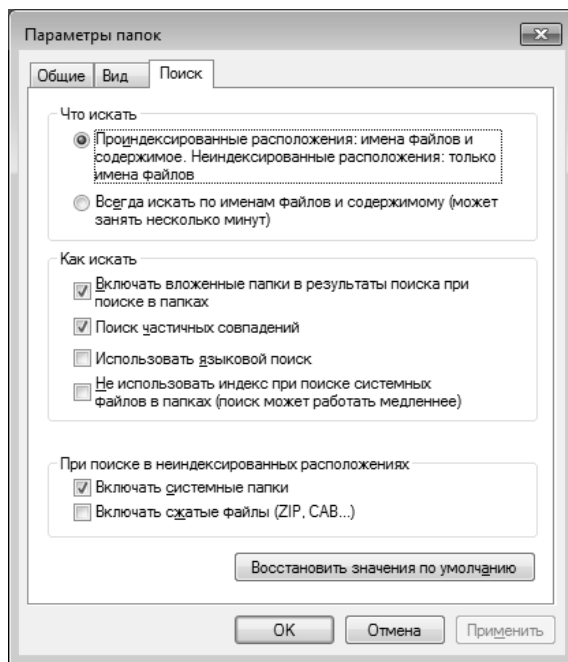


Рис. 6.15. Настройка общих параметров поиска

Операционная система способна выполнять поиск в папках по-разному: как контекстный или как файловый. Например, технология поиска может зависеть от того, проиндексирована или нет папка, в которой выполняется поиск. Соответствующие настройки выполняются на вкладке Поиск уже знакомого нам диалогового окна Параметры папок (Пуск ► Панель управления ► Оформление и персонализация ► Параметры папок), представленной на рис. 6.15.

Индексация данных

Служба поиска данных опирается на предварительное индексирование данных, в ходе которого просматриваются огромные массивы данных, и составляются указатели (индексы), с помощью которых поисковое задание выполняется очень быстро. Индексирование данных — очень трудоемкая и ресурсоемкая операция. Поэтому на практике ее назначают на периоды времени, когда компьютер наименее загружен.

По умолчанию операционная система выполняет индексацию только для следующих мест:

- ☐ Главное меню
- ☐ папки, входящие в состав библиотек;
- ☐ автономные файлы;
- ☐ папки Microsoft Outlook (если эта программа установлена на компьютере).

Для того чтобы содержимое папки было проиндексировано, проще всего добавить ее в состав одной из библиотек.

Если требуется индексировать данные, расположенные в других местах, следует выполнить дополнительную настройку в диалоговом окне Параметры индексирования. Оно открывается с помощью одноименного компонента в Панели управления (Пуск ► Панель управления ► Параметры индексирования).

Существует два основных метода управления индексацией данных:

- ☐ настройка области индексирования;
- ☐ настройка фильтров индексирования.

В первом случае формируется список папок, внутри которых должны индексироваться все объекты. Во втором случае формируется список типов файлов (по расширениям имен), подлежащих индексации.

6.6. Настройка шрифтов

Важным преимуществом графических операционных систем является возможность гибкого управления как экранными, так и печатными шрифтами в документах. Операционная система обеспечивает единство принципов применения шрифтов в самых разнообразных приложениях.

Растровые и векторные шрифты

Операционные системы Windows позволяют работать с двумя классами шрифтов: растровыми и векторными. Символы растровых шрифтов образуются как комбинации точек в матрице заданного размера. Достоинством растровых шрифтов является высокая скорость отображения символьных данных на экране. В связи с этим операционная система использует растровые шрифты в качестве экранных при отображении системной информации. Основным недостатком растровых шрифтов является негибкость управления размером и начертанием символов.

Размеры символов растровых шрифтов определяются размерами матрицы, на базе которой эти символы построены из комбинации точек. Характерные размеры: 8×12; 10×16; 13×22 и т. п. Изменение размера или начертания шрифта выполняется заменой символьного набора.

Символы векторных шрифтов представляют собой криволинейные контуры, составные элементы которых описываются математическими формулами. Это позволяет не хранить отдельно символьные наборы разных размеров. Управление размером (и некоторыми видами начертания) шрифта происходит программно. При отображении на экране или при выводе на печать символы любых размеров строятся из одного и того же символьного набора, поэтому векторные шрифты называют также масштабируемыми.

Векторные шрифты могут использоваться как в качестве экранных, так и в качестве печатных. Применение векторных шрифтов при подготовке документов позволяет реализовать принцип соответствия экранного изображения печатному — так называемый принцип WYSIWYG (What You See Is What You Get). В соответствии с этим принципом мы наблюдаем оформление документа на экране таким, каким оно будет при выводе с помощью печатающего устройства.

Типы векторных шрифтов

При реализации концепции векторных шрифтов возможны различные подходы к построению контуров из простейших кривых линий. В связи с этим существует несколько разных типов векторных шрифтов. Наиболее известными являются: Type 1 (PostScript), True Type и Open Type.

Стандарт Type 1 (PostScript) является более ранним. Он был разработан компанией Adobe Systems, лидером в разработке программного обеспечения для устройств печати и полиграфических систем. При работе с приложениями, выпущенными этой компанией, иногда целесообразно использовать векторные шрифты Type 1 (PostScript). Предыдущие версии операционной системы Windows не могли работать с ними напрямую и нуждались в специальной программе-посреднике. В частности, в качестве такой программы обычно применялись различные версии программы Adobe Type Manager.

Стандарт True Type, в свою очередь, был разработан компанией Microsoft Corporation для деловых приложений пакета Microsoft Office. Именно на него и были ориентированы все ранние операционные системы Windows до системы Windows XP (2001 г.).

Стандарт Open Type представляет собой компромиссное решение, достигнутое двумя мировыми лидерами программного обеспечения. С шрифтами этого типа должны работать приложения обеих корпораций. Современные операционные системы Windows XP, Windows Vista, Windows 7 изначально поддерживают все три стандарта, и использование любого типа шрифтов не вызывает проблем у современных приложений.

Файлы, содержащие данные о конструкции шрифтовых наборов, находятся в папке \Windows\Fonts, но эту папку не следует обслуживать напрямую. Гарантию надежной регистрации шрифтов в Реестре операционной системы дают специальные средства обслуживания.

Общее средство работы со всеми категориями векторных и растровых шрифтов находится на Панели управления (Пуск ► Панель управления ► Оформление и персонализация ► Шрифты) (рис. 6.16).

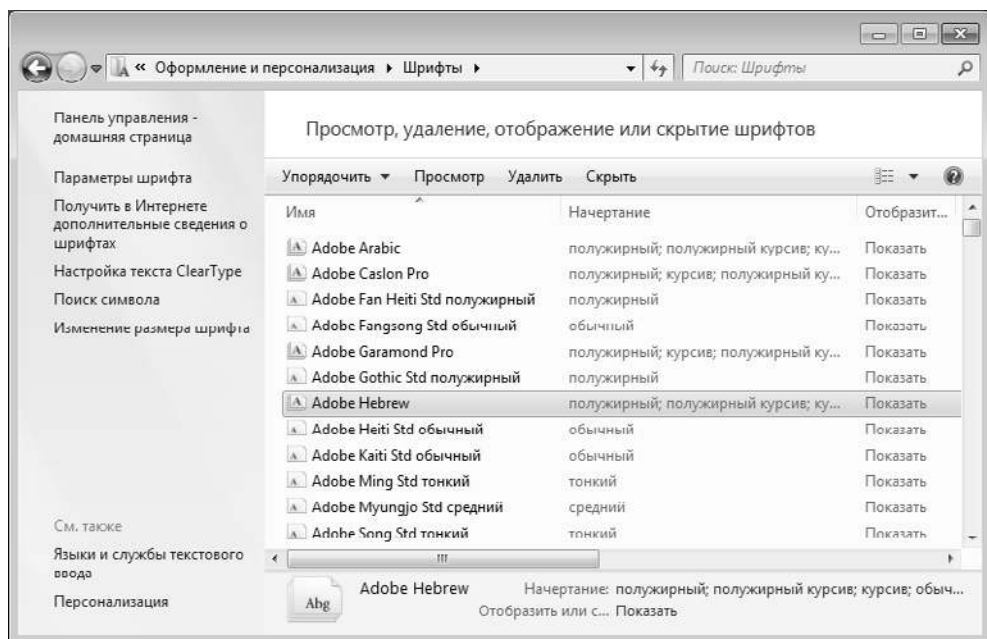


Рис. 6.16. Системный компонент для работы со шрифтами

Установка и удаление шрифтов

Перед установкой нового шрифта рекомендуется закрыть все работающие приложения. Для установки шрифта можно воспользоваться командой Установить контекстного меню файла шрифта. Также можно просто перетащить файл шрифта на страницу Шрифты панели управления.

Удаление шрифтов производится командой Удалить панели инструментов страницы Шрифты. Соответствующие шрифты при этом должны быть выделены.

6.7. Прочие настройки Windows

Настройка системных часов и системного календаря

При сохранении любого файла вместе с ним сохраняются данные о дате и времени создания или последнего изменения. Это происходит в полном соответствии с текущими настройками системных часов и системного календаря компьютера.

Средства настройки часов и календаря находятся ниже уровня операционной системы. Они относятся к базовому программному обеспечению компьютера и располагаются в его базовой системе ввода и вывода (BIOS). Сама по себе операционная система Windows не ведет учет времени, но, опираясь на показания системных часов, она обеспечивает следующие функциональные возможности:

- ☐ сохраняет показания системных часов вместе с атрибутами файлов при каждой операции сохранения данных;
- ☐ предоставляет пользователю для настройки системных часов и календаря более удобный интерфейс, чем тот, который предоставляет система BIOS;
- ☐ ведет автоматический учет таких факторов, как изменение поясного времени (это важно для портативных компьютеров), переход на «летнее» и «зимнее» время;
- ☐ обеспечивает корректный учет последних цифр года при смене века;
- ☐ обеспечивает возможность корректировки показаний системных часов через Интернет.

Настройку системных часов и системного календаря выполняют в диалоговом окне **Дата и время**, которое открывают с помощью соответствующего значка **Панели управления** или из контекстного меню индикатора времени, расположенного на панели индикации (рис. 6.17).

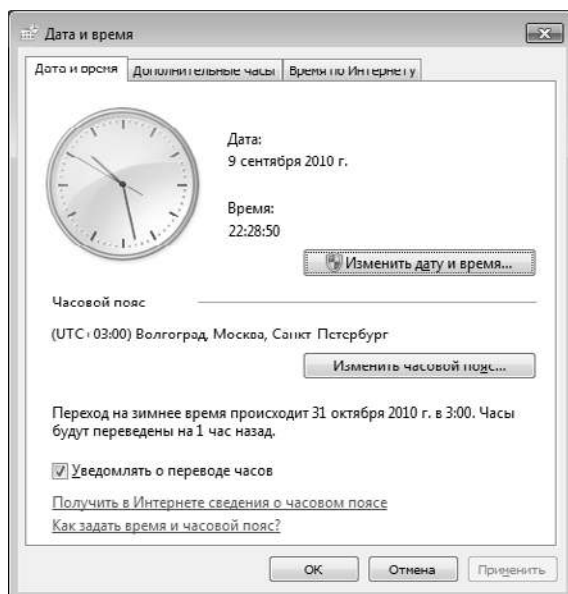


Рис. 6.17. Настройка параметров даты и времени

Год, месяц, дату и время устанавливают в окне Настройка времени и даты, которое открывается командной кнопкой Изменить дату и время. В свою очередь, часовой пояс задают в диалоговом окне Выбор часового пояса, которое открывают кнопкой Изменить часовой пояс.

Вкладка Дополнительные часы позволяет настраивать отображение дополнительных часов для других часовых поясов.

Вкладка Время по Интернету позволяет автоматически установить на компьютере точное время при наличии подключения к Интернету. Синхронизацию часов обеспечивает специальная служба Интернета. После первичной настройки параметров этой вкладки операционная система регулярно обращается на соответствующий сервер и поддерживает правильность показаний системных часов.

Настройка национальных стандартов и форматов

Операционная система Windows в значительной степени учитывает национальные различия, связанные с форматами записи чисел, дат, времени, денежных сумм и т. п. Так, например, в России принято представлять даты в формате дд.мм.гг (день-месяц-год), а в США — мм.дд.гг (месяц-день-год). Например, запись 08.03.10 в России означает 8 марта 2010 года. Та же запись в США означает 3 августа того же года.

Другой пример: в России десятичная часть дробного числа отделяется от целой части с помощью запятой, а в США — точкой. Есть различия в форматах записи времени, отрицательных чисел, денежных сумм, единиц измерения физических и денежных величин. Подобные различия касаются не только России и США, но и других стран.

Обычно при установке локализованной версии операционной системы настройки национальных стандартов и форматов производится автоматически, согласно тому, как указана страна пребывания в устанавливающей программе. Однако при работе с некоторыми приложениями, не адаптированными к использованию в конкретной стране, необходимо редактировать настройки, принятые по умолчанию. Для этого служат средства вкладок диалогового окна Язык и региональные стандарты (Пуск ► Панель управления ► Часы, язык и регион).

На вкладке Языки и клавиатуры этого окна имеются кнопки Изменить клавиатуру и Установить или удалить язык. Первая открывает доступ к средствам для установки дополнительных раскладок клавиатуры и выбора клавиатурной комбинации, переключающей раскладку. Вторая — для установки дополнительных языковых пакетов.

Практическое занятие

Упражнение 6.1. Настройка свойств мыши

Отработка приемов настройки графического манипулятора. Гармонизация настроек чувствительности манипулятора к перемещению и к двойному нажатию основной кнопки.

1. Откройте диалоговое окно Свойства: Мышь (Пуск ► Панель управления ► Оборудование и звук ► Мышь).

2. Щелкните дважды в области проверки на панели **Скорость выполнения** двойного щелчка. Убедитесь, что при двойном щелчке элемент срабатывает, а при двух отдельных щелчках с интервалом 0,3...0,5 с — срабатывания нет.
3. Методом перетаскивания переместите движок **Скорость** в крайнее правое положение. Убедитесь, что при этом интервал времени между двумя отдельными щелчками, составляющими двойной щелчок, чрезмерно занижен и выполнить двойной щелчок очень трудно.
4. Переместите движок в крайнее левое положение и убедитесь в том, что два отдельных щелчка интерпретируются как двойной щелчок.
5. Экспериментально выберите наиболее удобное для себя положение движка.
6. Откройте вкладку **Параметры указателя**.
7. Уменьшите чувствительность мыши, переместив движок **Задайте скорость движения указателя** в крайнее левое положение. Щелкните на кнопке **Применить**.
8. Установите указатель мыши примерно в центре экрана. Не отрывая запястья от поверхности стола, подвигайте мышь в направлении влево-вниз — вправо-вверх. Убедитесь в том, что указатель мыши не достигает левого нижнего и правого верхнего углов экрана.
9. Переместите движок **Задайте скорость движения указателя** в крайнее правое положение. Щелкните на кнопке **Применить**.
10. Убедитесь в том, что указатель мыши можно провести от левого нижнего до правого верхнего угла экрана, не отрывая запястье от поверхности стола.
11. Экспериментально выберите наиболее удобное для себя положение движка. После каждого изменения его положения не забывайте задействовать командную кнопку **Применить**. Оптимальный выбор может зависеть от конкретной модели мыши, наличия свободного места на поверхности стола и привычных навыков работы.
12. Закройте диалоговое окно **Свойства: Мышь**.

Итоги. Мы научились настраивать чувствительность графического манипулятора. Необходимые для этого средства предоставляет компонент **Панели управления** под названием **Мышь**.

Упражнение 6.2. Настройка оформления Рабочего стола

Отработка приемов изменения фонового рисунка Рабочего стола на примере рисунков, входящих в стандартную поставку операционной системы Windows Vista.

1. Включите компьютер, дождитесь окончания загрузки операционной системы. Щелкните правой кнопкой мыши на свободном от значков участке Рабочего стола.
2. Выберите в контекстном меню пункт **Персонализация** — откроется окно компонента **Персонализация**.

3. Активируйте ссылку Фон Рабочего стола — откроется компонент Фоновый рисунок рабочего стола.
4. В раскрывающемся списке Расположение изображения выберите пункт Фоны рабочего стола Windows.
5. Выберите один из рисунков. Зафиксируйте имя файла, в котором данный рисунок хранится, например img10.jpg.
6. Выберите способ размещения фонового рисунка на Рабочем столе. Возможные варианты: Заполнение, По размеру, Растянуть, Замостить и По центру.
7. Закройте компонент Фоновый рисунок рабочего стола кнопкой ОК.
8. Сверните все открытые окна и убедитесь, что фоновый рисунок Рабочего стола изменился.

Итоги. Мы научились изменять фоновый рисунок Рабочего стола. Необходимые для этого средства предоставляет компонент Персонализация. Его можно открыть, используя контекстное меню Рабочего стола.

Упражнение 6.3. Выполнение файлового поиска

С помощью механизма файлового поиска установить, в какой папке операционная система хранит стандартные фоновые рисунки.

1. Нажатием кнопки Пуск откройте окно Компьютер.
2. В поле поиска введите имя разыскиваемого файла. Например, img10.jpg. При поиске можно использовать подстановочные символы, например img*.* или, скажем, img??.jpg.
3. На панели результатов поиска наблюдайте появление ссылки на разыскиваемый объект. Чтобы установить место размещения найденного объекта, следует включить представление найденных результатов в виде таблицы. В частности, искомый файл img33.jpg располагается в папке C:\Windows\Web\Wallpaper.

Итоги. Мы научились выполнять файловый поиск и с его помощью установили, что операционная система Windows 7 хранит файлы фоновых рисунков в папке C:\Windows\Web\Wallpaper.

Упражнение 6.4. Автоматический запуск приложений

Практическая отработка приемов автоматического запуска приложений.

1. Включите персональный компьютер и дождитесь окончания загрузки операционной системы.
2. Запустите программу Проводник (Пуск ► Все программы ► Стандартные ► Проводник).
3. На левой панели Проводника разыщите папку \Главное меню\Автозагрузка. У каждого пользователя эта папка своя. К ней ведет следующий путь:

C:\Пользователи\{Имя пользователя}\AppData\Roaming\Microsoft\Windows\Главное меню\Программы\Автозагрузка\

Существует удобный прием для быстрого доступа к личной папке \Автозагрузка. Нужно пройти по пути Пуск ► Все программы, найти программную группу Автозагрузка, вызвать для нее контекстное меню нажатием правой кнопки мыши и выбрать в меню команду Открыть.

4. На левой панели Проводника раскройте папку C:\Windows\System32. На правой панели разыщите значок программы Калькулятор (Calc.exe). В случае необходимости используйте полосы прокрутки. Если есть трудности с розыском объекта Calc.exe, включите режим сортировки объектов по имени.
5. Методом специального перетаскивания (при нажатой правой кнопке мыши) перетащите значок приложения Calc.exe с правой панели Проводника на Рабочий стол. В открывшемся меню специального перетаскивания выберите пункт Создать ярлык. Вызовите контекстное меню для только что созданного ярлыка и выберите в нем команду Вырезать.
6. Откройте личную папку \Автозагрузка. Выполните в ней команду контекстного меню Вставить. Убедитесь в том, что в папке появился ярлык программы Калькулятор.
7. Завершите работу с операционной системой и выключите компьютер.
8. Включите компьютер, дождитесь окончания загрузки операционной системы и убедитесь в том, что произошел автоматический запуск программы Калькулятор.
9. Любым способом откройте окно личной папки \Автозагрузка и удалите ярлык Калькулятор.

Итоги. Мы научились находить личную папку \Автозагрузка с помощью Проводника Windows и использовать ее для автоматической загрузки приложений при запуске операционной системы.

Самостоятельная работа

Задание. Применение справочной системы при решении проблемных вопросов

Операционная система Windows 7 обеспечивает возможность безопасной работы на одном компьютере нескольких пользователей. О том, как выполняется регистрация пользователя, его подключение, защита конфиденциальной информации и смена пользователей, можно узнать с помощью справочной системы Windows 7.

1. Запустите справочную систему (Пуск ► Справка и поддержка).
2. С помощью средств поиска разыщите статьи, в которых рассмотрены вопросы работы нескольких пользователей за одним компьютером.
3. Используя первичные результаты поиска, постарайтесь найти ключевые слова, обеспечивающие наиболее эффективный поиск.
4. Подготовьте краткий отчет о принципах использования одного компьютера несколькими пользователями.

Глава седьмая

Стандартные приложения Windows

Основное назначение операционных систем — обеспечение взаимодействия человека, оборудования и программ. От операционных систем не требуется наличия средств, предназначенных для исполнения конкретных прикладных задач, — для этого есть прикладное программное обеспечение. Тем не менее в операционную систему Windows 7 входит ограниченный набор прикладных программ, с помощью которых можно решать некоторые простейшие повседневные задачи, пока на компьютере не установлены более мощные программные средства. Такие программы, входящие в поставку Windows, называют *стандартными приложениями*. В силу особой простоты их принято также рассматривать в качестве учебных. Знание приемов работы со стандартными приложениями позволяет ускорить освоение специализированных программных средств.

7.1. Стандартные прикладные программы

Программа Блокнот

Блокнот — это простейший текстовый редактор, который можно также использовать в качестве удобного *средства просмотра* текстовых файлов (формат .TXT, .HTML, .XML и некоторые другие). Для создания текстовых документов его применяют редко (только для небольших записок), но данную программу удобно использовать для отработки навыков работы с клавиатурой. Программа запускается командой Пуск ► Все программы ► Стандартные ► Блокнот. Пример ее рабочего окна показан на рис. 7.1.

На примере программы Блокнот мы познакомимся с некоторыми приемами создания, редактирования и сохранения документов, типичными для большинства приложений Windows.

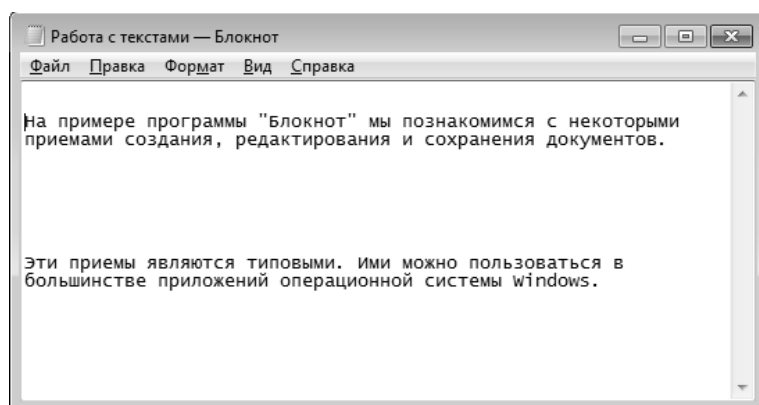


Рис. 7.1. Окно программы Блокнот

Ввод текста с помощью клавиатуры

Текст вводят с помощью алфавитно-цифровых клавиш. Для ввода прописных букв используют клавишу **SHIFT**. Если нужно ввести длинный ряд (поток) прописных символов, клавиатуру можно переключить с помощью клавиши **CAPS LOCK**.

Когда текст достигает правой границы окна, он может автоматически перетекать на новую строку, но может продолжаться далее, пока не будет нажата клавиша **ENTER**. Чтобы включить (или отключить) режим автоматического перетекания текста, используют команду **Формат ▶ Перенос по словам**.

Понятие курсора

Место документа, в которое происходит ввод текста (*точка ввода*), отмечается на экране вертикальной чертой, которую называют *курсором*. Не надо путать курсор с указателем мыши — это два разных понятия. Указатель мыши — это активный элемент управления, а курсор — это только маркер, не выходящий за пределы документа.

В прошлом, до появления графических операционных систем, указатель мыши называли курсором, но сегодня эти понятия различают. В редакторе Блокнот нетрудно убедиться в том, что, когда курсор находится в одном месте в тексте документа, указатель мыши можно свободно перемещать по полю документа и даже вне окна программы.

Переключение между русскими и латинскими символами

При наборе текста иногда приходится переключаться между русскими и латинскими символами. Это делается общесистемным способом, то есть метод переключения между символьными наборами не зависит от конкретной программы, а выполняется во всех программах одинаково. Это функция операционной системы.

Для того чтобы узнать, какой комбинацией клавиш на данном компьютере выполняется переключение раскладок клавиатуры, надо посмотреть, как настроены свойства языка (**Пуск ▶ Панель управления ▶ Часы, язык и регион ▶ Язык и региональные стандарты**).

Выбрав вкладку **Языки и клавиатуры**, нажмите кнопку **Изменить клавиатуру** — откроется диалоговое окно **Языки и службы текстового ввода** (рис. 7.2). В этом окне откройте вкладку **Переключение клавиатуры** и выясните, какое сочетание клавиш используется в качестве переключателя языков при вводе текста.

Обычно для переключения между русской и английской раскладками используют комбинацию клавиш **ALT+SHIFT**. Если включено отображение языковой панели на Панели задач, то индикатор текущего языка отображается рядом с **Панелью индикации**. В этом случае для переключения между языками достаточно щелкнуть мышью на данном индикаторе.

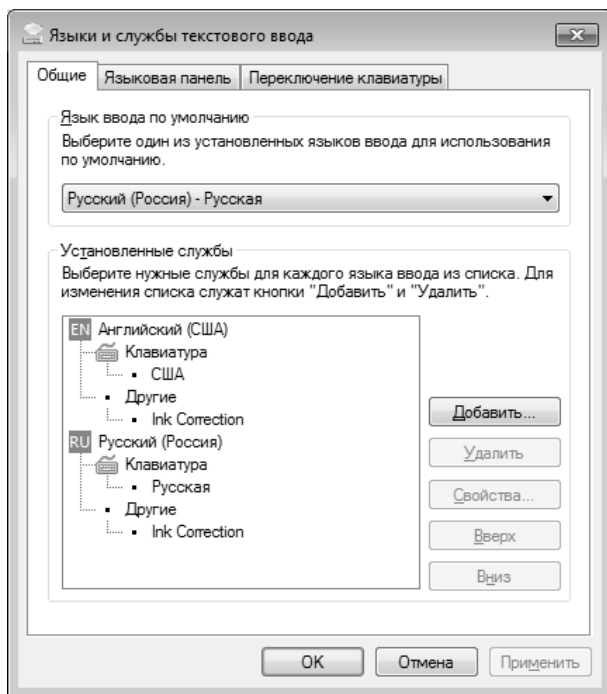


Рис. 7.2. Средство управления языками ввода и раскладками клавиатуры

Выбор шрифта

Размер и форма символов языка определяются использованным шрифтом. Редактор Блокнот слишком прост для того, чтобы позволить использование разных шрифтов в документе, но выбрать один шрифт, используемый для отображения документа, он позволяет. Это выполняется командой **Формат ► Шрифт**, которая открывает системное диалоговое окно **Выбор шрифта**, представленное на рис. 7.3.

В списке **Шрифт** можно выбрать один из возможных шрифтов. Здесь представлены все шрифты, установленные на компьютере. Не все шрифтовые наборы могут иметь в своем составе символы русского языка, поэтому при выборе шрифта требуется либо предварительное знание, либо свободное экспериментирование.

В списке **Начертание** можно задать начертание для избранного шрифта. Обычно используют четыре основных типа начертания: прямое светлое (обычное), *наклонное (курсив)*, **полужирное** и **полужирный курсив**. Выбор начертания, как и выбор шрифта, относится только к способу отображения документа (в более мощных текстовых редакторах и процессорах в одном документе можно применять разные шрифты и разные начертания).

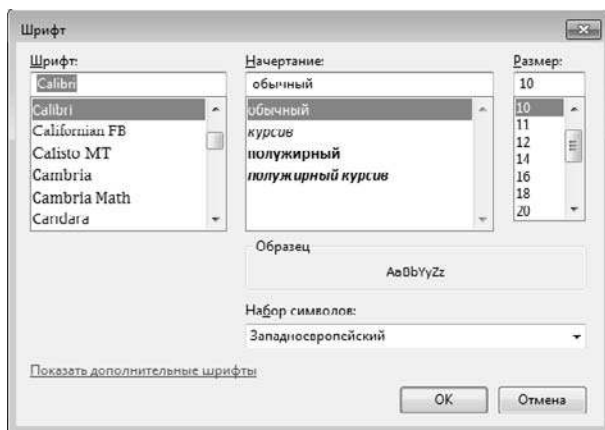


Рис. 7.3. Выбор шрифта в программе Блокнот

В списке **Размер** выбирают размер шрифта. Размеры шрифтов измеряются в *пунктах*. Пункт — это типографская единица измерения, равная 1/72 дюйма (0,353 мм). Для того чтобы документ хорошо читался на экране, обычно используют шрифт размером 12 пунктов.

Сохранение созданного документа

Созданный документ сохраняют на жестком или гибком магнитном диске в виде нового файла. При сохранении следует указать имя файла. Программа Блокнот не осуществит сохранение, пока имя не задано. Для сохранения нового документа служит команда **Файл** ► **Сохранить как**. По этой команде открывается диалоговое окно **Сохранение**, представленное на рис. 7.4.

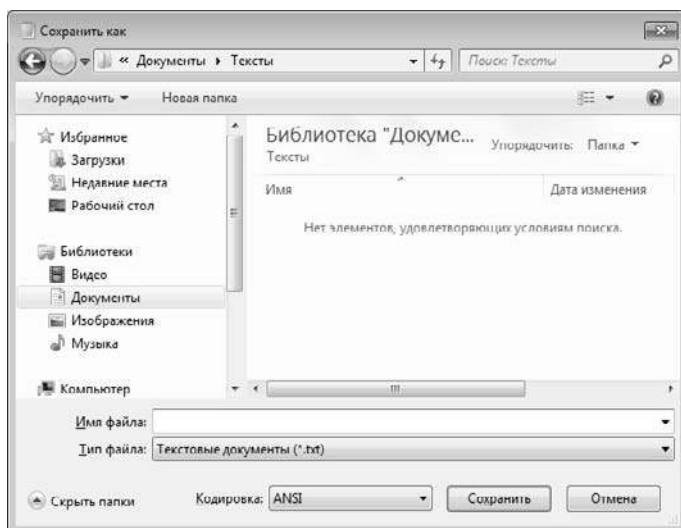


Рис. 7.4. Сохранение файла документа

В этом окне выбирают папку, в которую будет сохраняться файл, и дают ему имя. Приемы сохранения файлов одинаковы для всех приложений Windows. Освоив их один раз, далее можно пользоваться ими в любых программах. В качестве папки, в которую редактор Блокнот сохраняет документы *по умолчанию*, служит папка \Документы. Большинство приложений Windows предлагают по умолчанию использовать для сохранения документов именно эту папку. В ней можно создать несколько папок для раздельного хранения документов, относящихся к разным темам (проектам). Папка \Документы удобна еще и тем, что если с одним компьютером работают несколько человек и при запуске операционной системы каждый пользователь проходит регистрацию, то система создает каждому свою особую папку \Документы, чтобы документы разных людей не перемешивались между собой.

Если предложенная папка \Документы соответствует желанию автора, то остается только ввести имя файла в поле **Имя файла** и щелкнуть на кнопке **Сохранить**. Если в этой папке нужно создать новую папку, надо использовать кнопку **Создание новой папки** и дать новой папке содержательное имя.

Если же для сохранения документа надо использовать произвольную папку, отличную от папки \Документы, ее надо разыскать. К некоторым папкам имеется удобный доступ через панель, расположенную в левой части окна. Особого внимания заслуживает значок **Недавние места**. С его помощью легко перейти в любую папку, к которой недавно уже обращались.

Приемы редактирования документов

Под *редактированием* понимают изменение уже существующих документов. Редактирование начинают с загрузки (открытия) документа. Для этого служит команда **Файл ► Открыть**. По этой команде на экране появляется стандартное диалоговое окно **Открыть**.

Для редактирования текстовых документов следует научиться управлять курсором. Его перемещают с помощью специальных *клавиш управления курсором*. Для перемещения курсора на экранную страницу вверх или вниз используют клавиши **PAGE UP** и **PAGE DOWN**. Для перевода курсора в начало текущей строки используют клавишу **HOME**, а в конец строки — клавишу **END**. В большинстве приложений Windows работают также комбинации клавиш **CTRL+HOME** и **CTRL+END**, переводящие курсор в начало или конец документа соответственно. Для произвольного размещения курсора используют указатель мыши — курсор устанавливается в место щелчка в рабочей области.

Удаление ошибочных символов выполняют клавишами **BACKSPACE** или **DELETE**. Разница между ними состоит в том, что первая удаляет символы, стоящие слева от курсора, а вторая — справа. Для удаления большого блока текста пользоваться клавишами редактирования неудобно. В таких случаях сначала выделяют текстовый блок, а потом нажимают клавишу **DELETE** (один раз). При этом удаляется весь выделенный блок.

Выделение больших блоков производят методом протягивания мыши. В этом случае для удаления удобно использовать команду **Удалить** контекстного меню. Существует и прием выделения текстовых фрагментов с помощью клавиатуры. Он выполняется клавишами управления курсором при нажатой клавише **SHIFT**.

Выделенные фрагменты текста можно не только удалять, но и копировать или перемещать. Эти приемы очень часто применяются при редактировании. Копирование и перемещение происходит через буфер обмена Windows. Напомним комбинации клавиш, которые следует запомнить:

- ☐ CTRL+C — копировать в буфер;
- ☐ CTRL+X — вырезать в буфер;
- ☐ CTRL+V — вставить из буфера.

Программа Блокнот не позволяет работать более чем с одним документом, но ее можно запустить два и более раз. В этом случае на экране можно иметь несколько окон программы с разными документами. Поставив такой эксперимент, нетрудно убедиться, что перенос текстовых фрагментов через буфер обмена возможен не только внутри одного окна, но и между окнами.

Сохранение отредактированного документа

Сохранение документа, прошедшего редактирование, отличается от сохранения нового документа хотя бы тем, что файл этого документа уже существует и не надо выбирать папку и давать файлу имя. Для его сохранения достаточно дать команду **Файл ▶ Сохранить**, и новая копия документа заместит старую. Однако бывают случаи, когда старую копию документа не следует замещать. В этом случае документ сохраняют либо в другую папку, либо под другим именем. В этом случае порядок действий тот же, что и при сохранении нового документа командой **Сохранить как**.

Средства автоматизации

Программа Блокнот слишком проста, чтобы иметь серьезные средства автоматизации. В более мощных текстовых редакторах и процессорах эти средства надо изучать специально, поскольку от них зависит эффективность работы. В этой же программе единственное средство автоматизации состоит в том, что при нажатии на клавишу F5 в документ автоматически в печатывается текущее время и дата. Это удобно для ведения деловых записей и дневников.

Графический редактор Paint

Графическими называют редакторы, предназначенные для создания и редактирования изображений (рисунков). Программа Paint — простейший графический редактор. По своим возможностям она не соответствует современным требованиям, но в силу простоты и доступности остается необходимым компонентом операционной системы. Не разобравшись с принципами управления этой программой, трудно осваивать другие, более мощные средства работы с графикой.

Программа запускается командой **Пуск ▶ Все программы ▶ Стандартные ▶ Paint**.

Основные понятия

Программа Paint является редактором *растровой* графики. Это важное замечание, поскольку кроме редакторов растровой графики существуют еще редакторы *векторной* графики. Приемы и методы работы с этими двумя различными классами

программ совершенно различны. В растровой графике мельчайшим элементом изображения является точка, которой на экране соответствует экранная точка (*пиксел*). Мельчайшим элементом векторной графики является линия, описываемая математическим выражением.

Рабочее окно программы Paint представлено на рис. 7.5.

В состав его элементов управления входят ленточные панели инструментов. На вкладке Главная собраны основные инструменты для работы с рисунком. Здесь можно выбрать инструмент для рисования, настроить его параметры. Вкладка Вид содержит инструменты, которые позволяют управлять отображением рисунка.

Кнопка Paint (расположена в левой верхней части окна, в строке заголовков ленты) открывает меню, которое позволяет сохранять изображения, печатать их, создавать новые.

Панель быстрого запуска (в верхней части окна) позволяет делать доступными наиболее часто используемые команды, выведя их кнопки.

Рабочая область программы занимает основную площадь окна.

Строка состояния находится в нижней части окна.

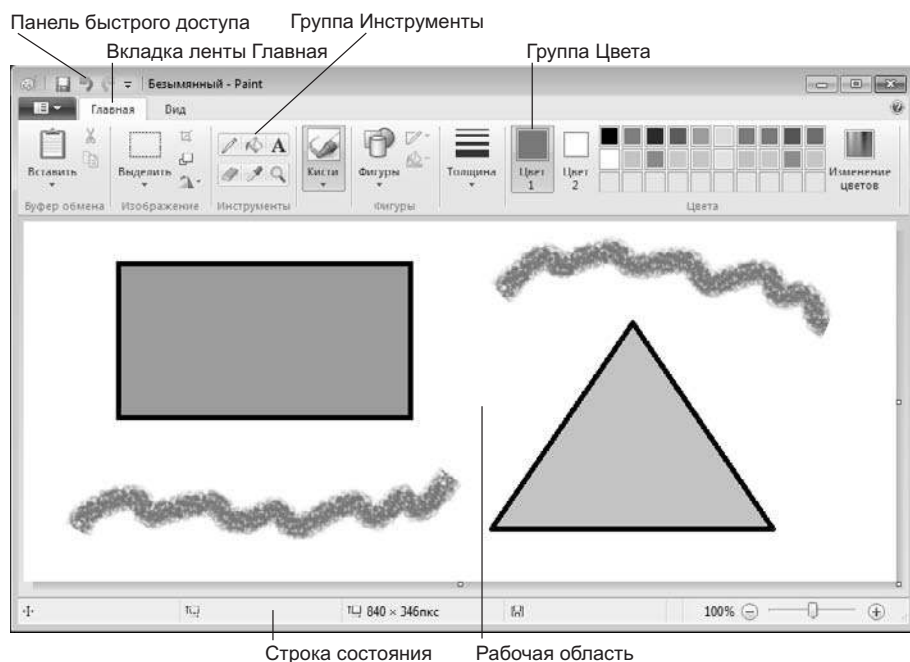


Рис. 7.5. Графический редактор Paint

Задание размера рабочей области

Перед началом работы следует хотя бы приблизительно задать размер будущего рисунка. Проще всего задать размер рисунка, перетаскивая мышью маркеры изменения размера, находящиеся по краям поля рисунка.

Так же размеры задают в полях **Ширина** и **Высота** диалогового окна **Свойства изображения** (Paint ► **Свойства**). До ввода размеров следует выбрать принятую единицу измерения с помощью одного из переключателей:

- ☐ Дюймы;
- ☐ См (сантиметры);
- ☐ Точки (пиксели).

В России не принято задавать размеры документов в дюймах. Размер в сантиметрах задают в тех случаях, когда предполагается вывод работы на печатающее устройство (принтер) или встраивание изображения на страницу с текстовым документом. В тех случаях, когда рисунок предназначен для воспроизведения на экране, в качестве единицы измерения выбирают Точки (пиксели). Так, например, если рисунок готовится для использования в качестве фона Рабочего стола, его размеры следует принять равными величине экранного разрешения монитора (640 × 480; 800 × 600; 1024 × 768 точек и т. д.).

Основные чертежно-графические инструменты

Все инструменты, кроме Ластика, выполняют рисование основным цветом (выбирается щелчком левой кнопки в палитре красок). Ластик стирает изображение, заменяя его фоновым цветом (выбирается щелчком правой кнопки мыши в палитре красок).

Инструмент **Линия** предназначен для вычерчивания прямых. Толщину линии выбирают в палитре настройки. Линии вычерчивают методом протягивания мыши. Чтобы линия получилась «строгой» (вертикальной, горизонтальной или наклонной под углом 45°), при ее вычерчивании следует держать нажатой клавишу **SHIFT**.

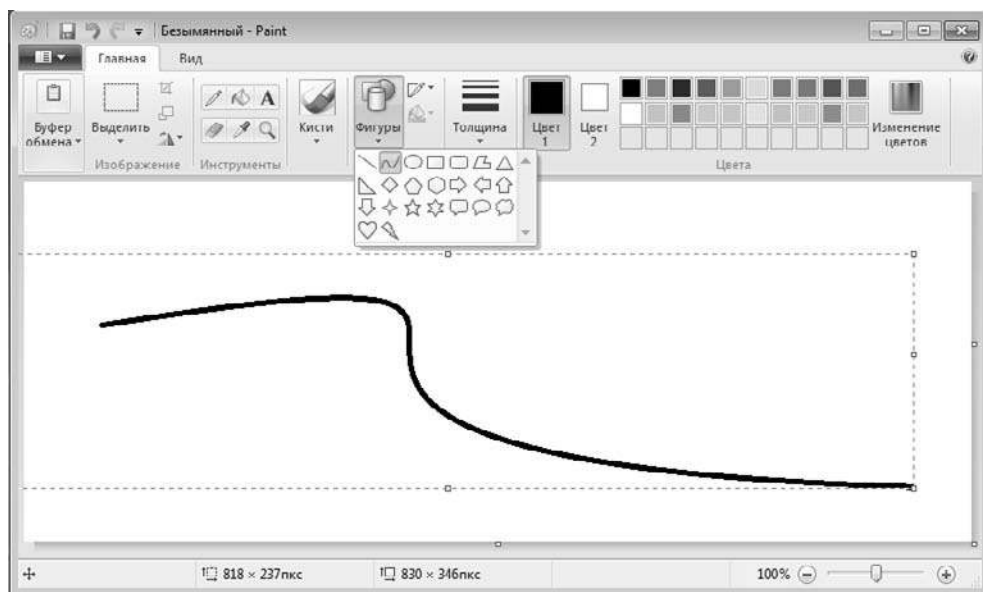
Инструмент **Карандаш** предназначен для рисования произвольных линий. Толщину линии выбирают в палитре настройки.

Инструмент **Кривая** служит для построения гладких кривых линий. Толщину предварительно выбирают в палитре настройки. Построение производится в три приема. Сначала методом протягивания проводят прямую линию, затем щелчком и протягиванием в стороне от линии задают первый и второй радиусы кривизны. Математически данная кривая, имеющая два радиуса кривизны и одну точку перегиба, является частным случаем кривой третьего порядка (*кривой Безье*) (рис. 7.6).

Инструмент **Кисть** можно использовать для свободного рисования произвольных кривых, как Карандаш, но чаще его используют для рисования методом *набивки*. Сначала выбирают форму кисти в палитре настройки, а потом щелчками левой кнопки мыши наносят оттиски на рисунок без протягивания мыши.

Инструмент **Распылитель** используют как для свободного рисования, так и для рисования методом набивки. Форму пятна выбирают в палитре настройки. При свободном рисовании вид рисунка зависит и от скорости движения указателя мыши.

Инструмент **Прямоугольник** применяют для рисования прямоугольных фигур. Рисование выполняется протягиванием мыши. В палитре настройки можно выбрать метод заполнения прямоугольника. Возможны три варианта: **Без заполнения** (рисуются только рамка), **Заполнение фоновым цветом** и **Заполнение основным цветом**.

**Рис. 7.6.** Рисование кривых

Если при создании прямоугольника держать нажатой клавишу **SHIFT**, образуется правильная фигура. Для прямоугольника правильной фигурой является квадрат.

Аналогичный инструмент **Скругленный прямоугольник** действует точно так же, но при этом получается прямоугольник со скругленными углами.

Инструмент **Многоугольник** предназначен для рисования произвольных многоугольников. Рисование выполняют серией последовательных щелчков с протягиванием. Если конечная точка многоугольника совпадает с начальной, то многоугольник считается замкнутым. Замкнутые фигуры могут автоматически заливаться краской в соответствии с вариантом заполнения, выбранным в палитре настройки.

Инструмент **Эллипс** служит для изображения эллипсов и окружностей. Окружность — это частный случай «правильного» эллипса. Она получается при рисовании с нажатой клавишей **SHIFT**.

Инструмент **Заливка** служит для заполнения замкнутых контуров основным или фоновым цветом. Заполнение основным цветом производится щелчком левой кнопки мыши, а заполнение фоновым цветом — щелчком правой кнопки. Если контур не замкнут, инструмент работает неправильно. В этом случае ошибочное действие надо немедленно отменить командой Панели быстрого доступа **Отменить** или комбинацией клавиш **CTRL+Z**.

Комбинацию **CTRL+Z** следует запомнить. Она отменяет последнее действие в большинстве приложений Windows и является удобным общесистемным приемом.

Инструмент **Выбор цветов** позволяет точно выбрать основной или дополнительный цвет не из палитры красок, а непосредственно из рисунка. Это важно,

когда надо обеспечить тождественность цвета в разных областях изображения. После выбора инструмента наводят указатель на участок рисунка с нужным цветом и щелкают кнопкой мыши. Если произошел щелчок левой кнопкой, текущий цвет становится основным, а если правой — фоновым.

Инструменты выделения областей

Два инструмента предназначены для работы с выделенными областями: **Выделение** и **Выделение произвольной области**. Действуют они одинаково, разница лишь в том, что инструмент **Выделение** формирует не произвольную, а прямоугольную выделенную область. С выделенной областью можно поступать так, как это принято во всех приложениях Windows: ее можно удалить клавишей **DELETE**, скопировать в буфер обмена (**CTRL+C**), вырезать в буфер обмена (**CTRL+X**) и вставить из буфера обмена (**CTRL+V**). Прием копирования и вставки выделенной области применяют для размножения повторяющихся фрагментов.

Масштабирование изображений

Для точной доводки рисунка иногда необходимо увеличить его масштаб. Максимальное увеличение — восьмикратное. Для изменения масштаба служат элементы управления группы **Масштаб** вкладки **Вид**.

В режиме четырехкратного или восьмикратного увеличения на рисунок можно наложить вспомогательную сетку (флажок **Показать сетку** на вкладке **Вид**). Каждая ячейка этой сетки представляет собой одну увеличенную точку изображения. В этом режиме удобно редактировать изображение по отдельным точкам. Сетку можно наложить и при более мелком масштабе, однако функций помощи при пиксельном редактировании изображений она выполнять не будет.

Трансформация изображений

Трансформациями называют автоматические изменения формы, расположения или размеров графических объектов. В программе Paint не слишком много инструментов трансформации, но все-таки они есть. Их можно найти в группе параметров **Изображение** на вкладке **Главная**.

Команды группы **Повернуть** позволяют поворачивать рисунок на 90° вправо и влево, а также на 180°. Здесь же можно найти команды для симметричного отображения рисунка относительно вертикальной или горизонтальной оси симметрии.

Кнопка **Изменить размер** открывает окно **Изменение размеров и наклона**. Его элементы управления позволяют растянуть рисунок по горизонтали и вертикали или наклонить относительно горизонтальной или вертикальной оси. Параметры растяжения задают в процентах или в пикселах, а параметры наклона — в угловых градусах (рис. 7.7).

Команда контекстного меню **Обратить цвета** действует как переключатель. При использовании этой команды цвет каждой точки изображения меняется на

«противоположный». В данном случае мы назвали «противоположным» тот цвет, который дополняет данный цвет до белого.

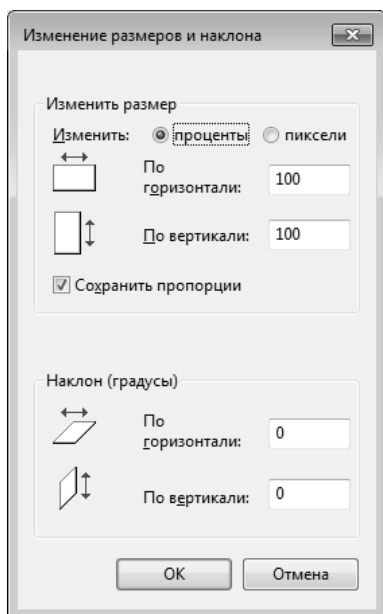


Рис. 7.7. Средство растягивания и наклона изображения

Ввод текста

Программа Paint — графический редактор и не предназначена для работы с текстом. Поэтому ввод текста в этой программе является исключением, а не правилом. Поскольку редактор относится к растровым, он строит изображение по точкам. Следовательно, текст после ввода станет «рисунком» и будет состоять из достаточно крупных точек растра. Поэтому избегайте использования мелких символов, которые смотрятся неопрятно. Рассматривайте режим работы с текстом в программе Paint только как средство для создания кратких и крупных заголовков.

Для ввода текста используют инструмент **Надпись**. Выбрав инструмент, щелкните на рисунке примерно там, где надпись должна начинаться, — на рисунке откроется поле ввода. В это поле вводится текст с клавиатуры.

При вводе текста в верхней части окна программы будет открыта новая вкладка — **Текст**. Она содержит элементы управления для настройки свойств текста (рис. 7.8).

О типе шрифта, его размере и начертании заботиться пока не надо — главное набрать текст без ошибок, а остальное все можно изменить позже. Размер поля ввода изменяют путем перетаскивания *маркеров области ввода* — небольших прямоугольных узлов, расположенных по сторонам и углам области ввода.

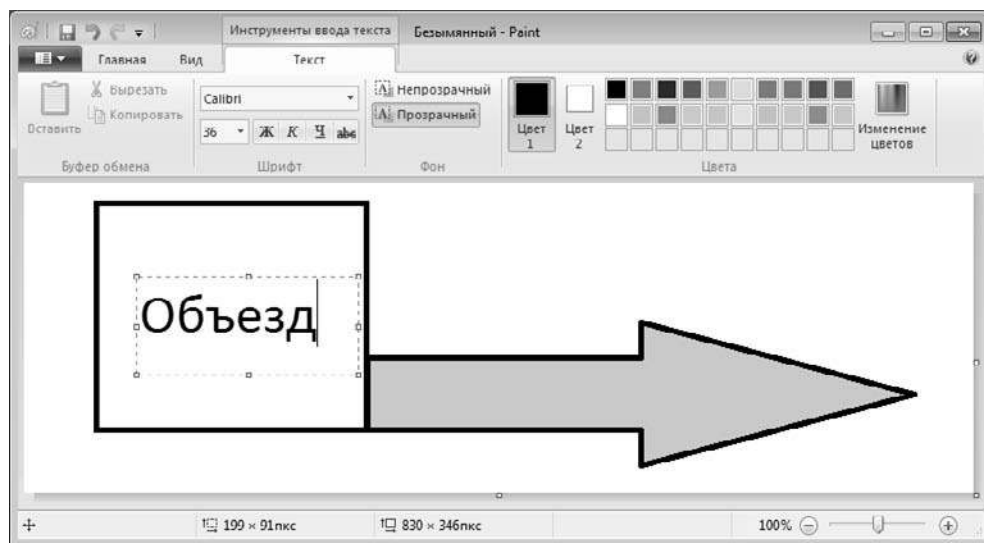


Рис. 7.8. Работа с текстом в Paint

О том, чего нет в редакторе Paint

В работе с вычислительной техникой безусловно важно знать возможности программных средств и приемы их использования. Но не менее важно знать и ограничения программных средств. Это позволяет двигаться вперед, осваивать новые продукты и приемы. Как мы уже говорили, графический редактор Paint — простейший, поэтому в нем нет многого из того, что есть в других современных графических редакторах.

1. *Автоматическое выделение областей.* Мы видели, как в редакторе Paint выполняется выделение прямоугольных и произвольных областей. В более мощных редакторах есть средства для автоматического выделения. Например, они могут работать по принципу подобия цвета: все элементы изображения, имеющие цвет, близкий к заданному, выделяются автоматически. Это позволяет точно выделять очень сложные контуры (операция называется *обтравкой контура*).
2. *Специальные методы заливки.* В программе Paint работают простейшая заливка одним цветом и несколько вариантов текстурной заливки. В более мощных программах обычно имеются также средства выполнения градиентной заливки (*градиентной заливкой* или *растяжкой* называют заливку с плавным переходом от одного цвета к другому) и множество вариантов текстурной заливки (при текстурной заливке замкнутый контур заполняется узором или рисунком, имитирующим фактуру материала, например дерева, металла, ткани и т. п.).
3. *Применение фильтров.* Фильтрами называют специальные методы автоматической обработки изображений или выделенного фрагмента. Например, с помощью фильтров можно управлять яркостью или контрастностью изображения. Существуют искажающие фильтры, например имитирующие просмотр рисунка

через стекло, смоченное водой и т. п. В редакторе Paint нет фильтров, но в других графических редакторах могут насчитываться десятки и сотни фильтров для создания специальных эффектов.

4. *Использование слоев.* В редакторе Paint мы работаем только с одним слоем изображения. Это не слишком удобно. В тех программах, где предусмотрена возможность создания слоев, можно разные объекты располагать на разных слоях, а потом объединять их. Слои могут быть прозрачными или полупрозрачными. С помощью слоев создают эффект туманной дымки на фотографиях или эффекты, когда объект как бы парит над поверхностью фона и отбрасывает тень на поверхность (особенно часто этот эффект применяют для создания «парящих надписей»).
5. *Трансформации.* На примере программы Paint мы познакомились с простейшими трансформациями изображения: *наклоном* и *растягиванием*. Существуют и более сложные трансформации, например *скручивание*. Особенно много трансформаций существует для преобразования трехмерных объектов.
6. *Использование подключаемых расширений (plug-ins).* Ни одна графическая программа не может содержать все мыслимые инструменты, фильтры, средства заливки и трансформации. Поэтому современные графические редакторы позволяют подключать дополнительные компоненты, называемые *расширениями*. Возможность модернизации программного обеспечения путем подключения дополнительных блоков, сделанных посторонними программистами, называют *принципом открытой программной архитектуры*. В последние годы этот принцип получил широкое развитие. Программы, обладающие открытой архитектурой, развиваются и совершенствуются быстрее, чем программы с *закрытой архитектурой*, модернизация которых посторонними лицами не предусмотрена.

Текстовый процессор WordPad

Текстовые процессоры, как и текстовые редакторы, служат для создания, редактирования и просмотра текстовых документов. Однако они выполняют еще одну важную функцию: форматирование документов. Под форматированием понимают приведение внешнего вида документа к заданной форме, в частности к формату печатного листа. Основными приемами форматирования являются: оформление документов несколькими шрифтовыми наборами, управление длиной строки и выравниванием текста в соседних строках, встраивание в текстовый документ информационных объектов иной природы, например рисунков.

В стандартную поставку операционных систем Windows входит текстовый процессор WordPad, который фактически является «облегченной» версией гораздо более мощной программы — Microsoft Word. На примере этой программы мы ознакомимся с простейшими приемами форматирования документов. Программа запускается командой Пуск ► Все программы ► Стандартные ► WordPad. Ее рабочее окно представлено на рис. 7.9. Как видно из этого рисунка, в отличие от текстового редактора Блокнот, окно текстового процессора содержит дополнительную панель элементов управления — панель форматирования.

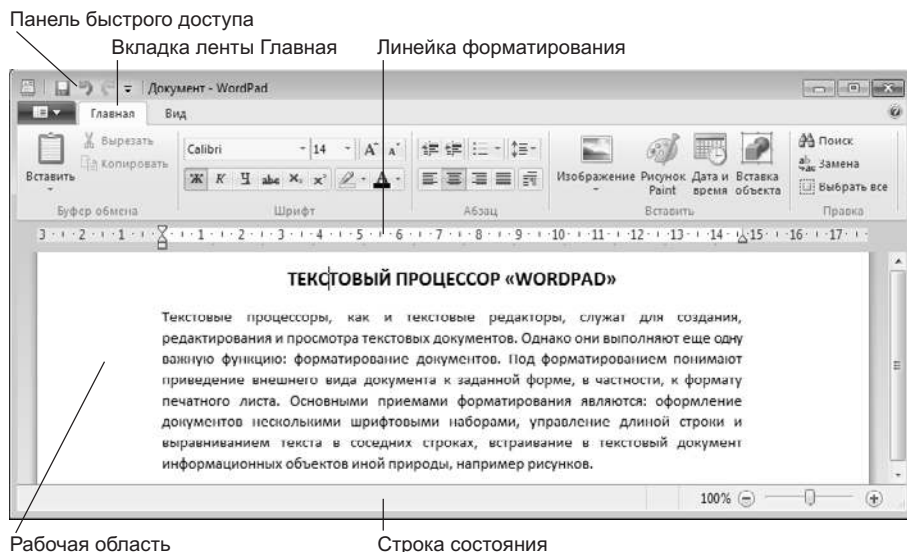


Рис. 7.9. Окно текстового процессора WordPad

Поскольку с приемами создания и редактирования документа мы знакомы по текстовому редактору Блокнот (см. выше), то на примере текстового процессора WordPad мы ознакомимся с простейшими приемами форматирования документов.

Настройка параметров печатной страницы

Форматирование документа предполагает получение полноценного бумажного оттиска на печатающем устройстве. Поэтому работа в текстовых процессорах начинается с задания параметров печатной страницы. Параметры страницы задают в диалоговом окне **Параметры страницы** (Wordpad ► **Параметры страницы**) (рис. 7.10).

Настройку параметров печатной страницы следует выполнять в соответствии с тем типом принтера, который предполагается использовать для печати. Для выбора принтера служит соответствующее поле в диалоговом окне **Печать** (Wordpad ► **Печать**). В раскрывающемся списке **Принтер** приведены те модели принтеров, на которые настроена конфигурация данного компьютера.

После выбора модели принтера выбирают параметры печатной страницы. Размеры листа бумаги выбирают в окне **Параметры страницы** в раскрывающемся списке **Размер**. В России в качестве стандартного машинописного листа принято использовать лист формата А4, имеющий размер 210 × 297 мм.

При печати принято ориентировать лист так, чтобы его высота была больше ширины (Книжная ориентация). Альбомная ориентация применяется в особых случаях, например при печати двух страниц на одном листе.

Печатное поле документа составляет не весь бумажный лист, поскольку со всех сторон документа должны оставаться белые поля. При выборе размеров полей необходимо учитывать следующие обстоятельства:

- ☐ если левое поле используют для брошюровки, оно должно иметь увеличенный размер;

- ❑ если при брошюровке предполагается обрезка блока, правое и нижнее поля должны иметь увеличенный размер;
- ❑ если при оформлении документа используются колонтитулы (верхние или нижние), для них следует предусмотреть увеличение размера соответствующих полей.

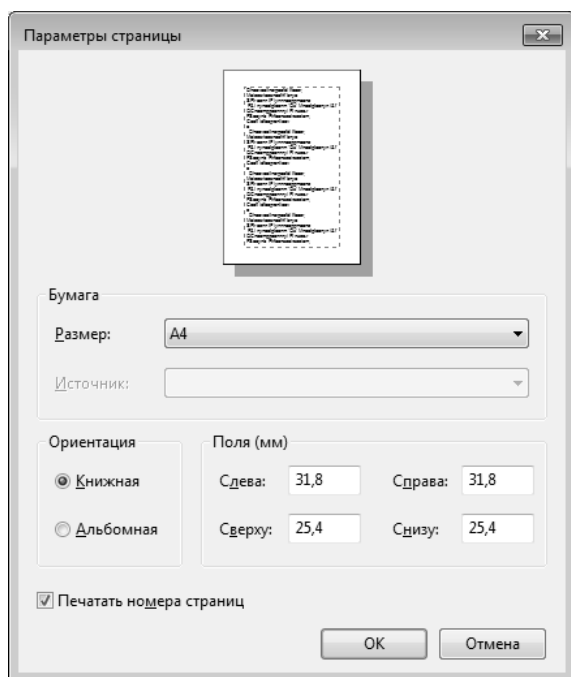


Рис. 7.10. Настройка параметров страницы в Wordpad

Конкретные значения размеров полей следует выяснить у заказчика документа (работодателя, администрации предприятия). Если никаких рекомендаций нет, можно задать для всех полей, кроме левого, по 15 мм, а для левого поля — 25 мм.

Настройка параметров абзаца

Абзац является минимальным элементом форматирования. Настройка параметров абзаца выполняется в диалоговом окне **Абзац**, которое можно открыть из контекстного меню абзаца, а также с помощью элементов управления группы **Абзац** вкладки **Главная**.

Здесь можно задать следующие параметры:

- ❑ величину отступа от левого поля;
- ❑ величину отступа от правого поля;
- ❑ величину специального отступа для первой строки абзаца (используется для создания «красной строки»);
- ❑ метод выравнивания: по левому полю, по центру, по правому полю, по ширине — так называется метод выравнивания, при котором текст выравнивается

и по левому, и по правому полям одновременно. Для большинства документов, написанных на русском языке, этот метод является стандартным.

Настройка параметров шрифтового набора

Тип используемого шрифта, его размер и начертание можно задать с помощью элементов управления, представленных на вкладке Главная. В отличие от редактора Блокнот текстовый процессор WordPad позволяет применять шрифтовое оформление как ко всему документу в целом, так и к отдельным, предварительно выделенным фрагментам.

Создание маркированных списков

Создание маркированных списков — характерная возможность большинства текстовых процессоров. В программе WordPad первая строка маркированного списка создается щелчком на кнопке Начать список на панели форматирования и выбором необходимого вида списка.

Последующие строки автоматически получают маркер после нажатия клавиши ENTER. Для прекращения маркировки надо просто повторить команду еще раз.

Управление табуляцией

Режим табуляции определяет характер линейного смещения текстового курсора в строке при последовательных нажатиях клавиши TAB. Табуляцией пользуются в тех случаях, когда есть необходимость оформления текста ровными столбцами, что в большинстве случаев необходимо при создании таблиц.

Позиции табуляции задают в диалоговом окне Табуляция (кнопка Вкладки в окне Абзац). Координаты позиции табуляции задаются в сантиметрах и измеряются от левого поля. Например, если задать три позиции (5 см, 10 см и 15 см), то при нажатии клавиши TAB текстовый курсор в зависимости от текущего положения смещается вправо к ближайшей позиции табуляции.

Поиск и замена текстовых фрагментов

Наличие средства поиска и замены текстового фрагмента — обязательный элемент текстовых процессоров. В программе WordPad средство поиска запускают кнопкой Поиск, которая расположена в группе параметров Правка вкладки Главная.

Текстовый фрагмент, подлежащий поиску, вводят в поле Что, а процесс поиска запускают щелчком на кнопке Найти далее. Установкой флажков Только слово целиком и С учетом регистра настраивают особенности поиска.

Поиск с одновременной заменой запускают кнопкой Замена из той же группы Правка вкладки Главная. Разыскиваемый фрагмент вводят в поле Что, а замещающий фрагмент — в поле Чем. Поиск выполняют командой Найти далее, замену фрагмента — командой Заменить, а глобальную замену по всему тексту — командой Заменить все.

Возможность автоматической замены используют для автоматизации ввода текста и редактирования. Так, например, если при вводе текста набирать слова в сокращенном виде: к-р, к-ра, к-ров и т. п., а потом по всему тексту заменить к-р на компьютер, то можно значительно сократить объем ввода с клавиатуры. Таким же образом правят систематические ошибки, обнаруженные в ходе редактирования.

7.2. Принципы внедрения и связывания объектов

Операционная система Windows позволяет:

- ☐ создавать комплексные документы, содержащие несколько разных типов данных;
- ☐ обеспечивать совместную работу нескольких приложений при подготовке одного документа;
- ☐ переносить и копировать объекты между приложениями.

Так, например, рисунок, созданный в графическом редакторе Paint, можно скопировать в текстовый документ, разрабатываемый в текстовом процессоре WordPad. То же можно делать и с фрагментами звукозаписи и видеозаписи. Разумеется, звуковой объект нельзя отобразить на печатной странице, но если документ электронный, то его можно вставить в текст в виде значка. Щелчок на этом значке во время просмотра документа позволит прослушать связанную с ним звукозапись.

Возможность использования в одном документе объектов различной природы является очень мощным инструментом Windows. Она основана на так называемой *концепции внедрения и связывания объектов* (OLE — Object Linking and Embedding).

Внедрение объектов

Под внедрением объектов подразумевается создание комплексного документа, содержащего два или более автономных объектов. Обычным средством внедрения объектов в документ является их *импорт* из готового файла, в котором данный объект хранится. Так, например, если в графическом редакторе Paint был создан и сохранен на диске файл рисунка ABCD.BMP, то в текстовом процессоре WordPad этот рисунок можно вставить в текстовый документ с помощью команды Главная ► Вставка ► Вставка объекта. При этом открывается диалоговое окно Вставка объекта.

Импорт вставляемого объекта обеспечивается переключателем Создать из файла, а его выбор на диске — кнопкой Обзор. Кроме графических объектов в текстовый документ можно внедрять и объекты другой природы — тексты, фрагменты звукозаписи и видеозаписи (рис. 7.11).

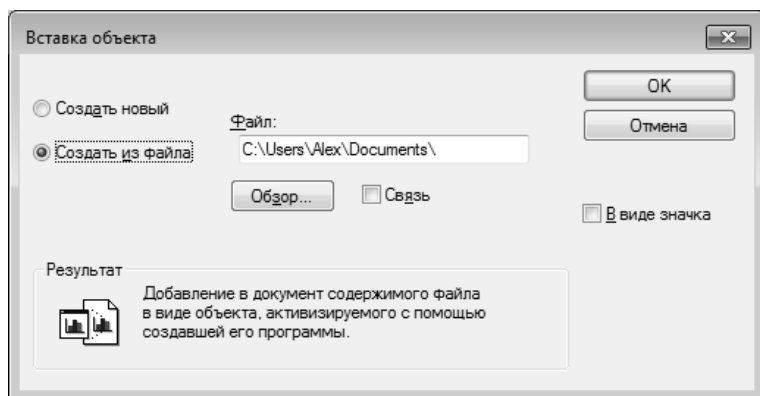


Рис. 7.11. Импорт графического объекта в текстовый документ

При сохранении комплексного документа происходит сохранение и текста, и всех внедренных в него объектов. Рисунок, ранее существовавший в виде отдельного графического файла, теперь внедрен в текстовый документ и располагается внутри него. Разумеется, при этом размер исходного текстового документа возрастает на величину внедренных объектов.

Связывание объектов

Однако мы могли поместить рисунок в текстовый документ и другим способом. В том же диалоговом окне **Вставка объекта** есть флажок, который называется **Связь**. Если установить этот флажок перед вставкой объекта, то происходит другой тип вставки, который называется *связыванием*. Связывание отличается от внедрения тем, что сам объект не вставляется в документ, а вместо этого вставляется только указатель на местоположение объекта. Когда при просмотре документа читатель дойдет до этого указателя, текстовый процессор обратится по адресу, имеющемуся в указателе, и отобразит рисунок в тексте документа.

При использовании связывания объектов, а не внедрения размер результирующего комплексного документа практически не увеличивается, так как указатель занимает очень мало места. Однако если не принять специальные меры, то при передаче такого документа заказчику не произойдет передача связанных объектов, поскольку они останутся в своих местах хранения. Это явление называется *разрывом*, или *потерей связи*. Потерянные связи надо восстанавливать. Потеря связи может происходить даже при простом перемещении связанных объектов из одной папки в другую. Таким образом, при использовании метода связывания объектов необходимо специально контролировать целостность связей между объектами и выполнять операции обслуживания этих связей (обновления и восстановления).

Сравнение методов внедрения и связывания

И тот и другой методы имеют свои области применения. Все зависит от формы и назначения документа. Внедряя объекты, мы избавляемся от необходимости поддерживать и обслуживать связи, но при этом можем получать файлы огромных размеров, с которыми трудно оперировать. Связывая объекты, мы резко уменьшаем размеры файлов и значительно повышаем производительность компьютера, но вынуждены следить за тем, чтобы все связанные объекты хранились строго в тех папках, в которые они были помещены в момент создания связи.

С принципами связывания и внедрения объектов непосредственно соприкасается принцип *совместного использования объектов*. В корпоративных вычислительных системах нередко используют стандартизированные объекты (бланки документов, логотипы предприятий и т. п.), доступ к которым (без права изменения) имеют большие группы сотрудников.

Такие объекты удобно вставлять в результирующий документ методом связывания. Во-первых, это позволяет значительно сократить объем документации предприятия, так как один и тот же объект может использоваться во всех документах без размножения. Во-вторых, такой подход позволяет администрации предприятия

легко изменять (в случае необходимости) стандартный объект и иметь уверенность в том, что при использовании любого документа, имеющего с ним связь, произойдет автоматическая подмена объекта. При таком подходе за пределы предприятия не выйдет ни один документ, напечатанный на устаревшем бланке, имеющем устаревшие реквизиты и т. п.

Итак, на практике обычно поступают следующим образом. Если документ готовится для печати на принтере или для просмотра на экране в пределах локальной сети предприятия, то объекты в него вставляют методом связывания. Если же документ готовится для передачи в электронном виде во внешние структуры, в него объекты внедряются.

OLE-серверы и OLE-клиенты

Объект — это очень специфическое образование, и не каждое приложение может его создать. Те приложения, которые способны создавать объекты для передачи другим приложениям, называются *OLE-серверами*, а те, которые позволяют внедрять или связывать чужие объекты в свои документы, называются *OLE-клиентами*. Например, при вставке рисунка в текстовый документ графический редактор играет роль OLE-сервера, а текстовый процессор — роль OLE-клиента.

7.3. Служебные приложения Windows

Служебные приложения Windows 7 предназначены для обслуживания персонального компьютера и самой операционной системы. Они позволяют находить и устранять дефекты файловой системы, оптимизировать настройки программного и аппаратного обеспечения, а также автоматизировать некоторые рутинные операции, связанные с обслуживанием компьютера.

В Главном меню служебные приложения Windows 7 сосредоточены в категориях Пуск ► Все программы ► Обслуживание и Пуск ► Все программы ► Стандартные ► Служебные. Они поставляются в составе операционной системы и устанавливаются вместе с ней (полностью или выборочно). Ниже приведена краткая характеристика основных служебных приложений.

Дефрагментация диска

Дефрагментация диска — служебное приложение, предназначенное для повышения эффективности работы жесткого диска путем устранения фрагментированности файловой структуры.

Наименьшей единицей хранения данных на диске является *кластер*. Если свободного места на диске достаточно, то файлы записываются так, что кластеры, в которые происходит запись, располагаются последовательно. В этом случае обращения к файлу происходят достаточно быстро, поскольку затраты времени на поиск очередных кластеров минимальны.

Если диск заполнен до отказа, запись на него возможна только после освобождения некоторого количества кластеров путем удаления файлов. При этом свобод-

ные области, образующиеся на диске, в общем случае не образуют одну большую непрерывную область. При попытке записать длинный файл на диск, имеющий прерывистую структуру свободных областей, файл делится на фрагменты, которые записываются туда, где для них нашлось место. Длительная работа с заполненным жестким диском приводит к постепенному увеличению фрагментированности файлов и значительному замедлению работы.

Программа Дефрагментация диска выполняет перекомпоновку файлов таким образом, что длинные файлы собираются из коротких фрагментов. В результате доступ к файлам заметно упрощается и эффективность работы компьютера возрастает.

Окно программы открывается через Главное меню командой Пуск ► Все программы ► Стандартные ► Служебные ► Дефрагментация диска (рис. 7.12). В этом окне можно настроить расписание для регулярного автоматического запуска программы или запустить процедуру дефрагментации накопителей немедленно.

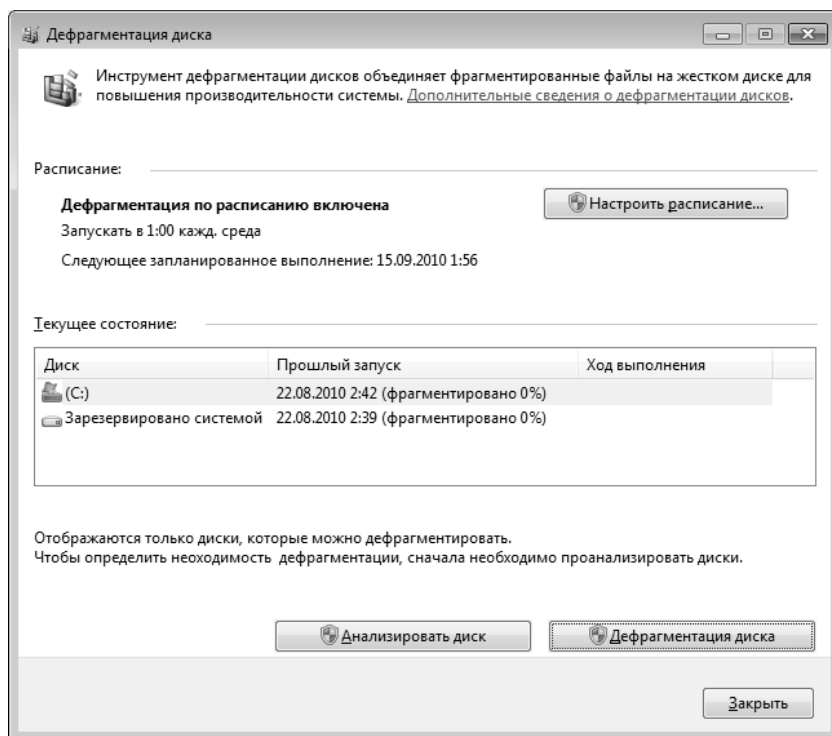


Рис. 7.12. Окно средства дефрагментации дисков

Сведения о системе

Сведения о системе — это специальный пакет программных средств, собирающих сведения о настройке операционной системы Windows 7, ее приложений и оборудования компьютерной системы. Средства этого пакета предназначены для специ-

алистов, выполняющих ремонтно-восстановительные работы. Его дополнительное преимущество состоит в том, что он позволяет провести диагностику компьютера с удаленного сервера.

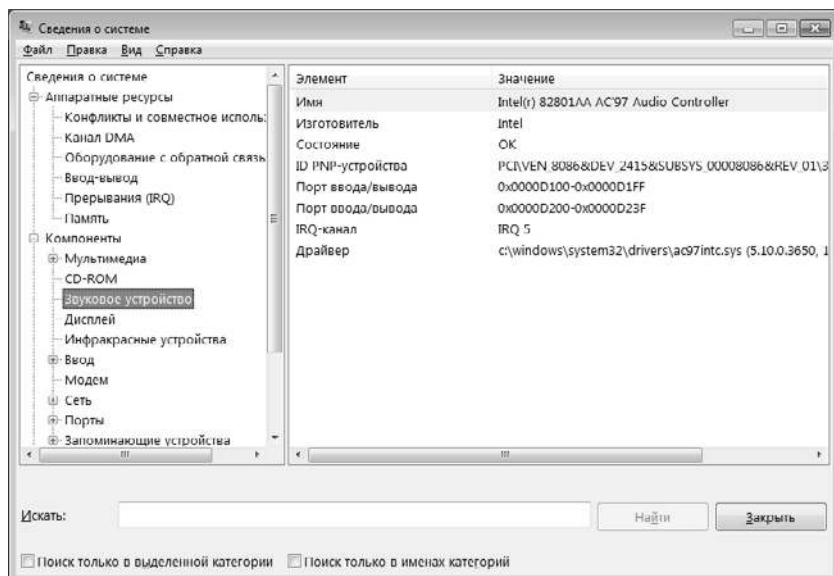


Рис. 7.13. Сведения о системе

Таблица символов

Кроме шрифтов с алфавитно-цифровыми символами в операционной системе Windows 7 можно использовать и специальные символьные наборы с дополнительными элементами оформления текстовых документов. В любом текстовом процессоре этими символьными наборами можно пользоваться точно так же, как обычными шрифтами. Однако если для обычных шрифтов раскладка клавиш понятна, то для символьных наборов нужны специальные средства, чтобы установить закрепление символов за клавишами клавиатуры.

Программа Таблица символов позволяет увидеть на экране все символы заданного набора и установить, какой символ какой клавише соответствует. Рабочее окно программы Таблица символов показано на рис. 7.14. В качестве примера в нее загружен символьный набор Wingdings, входящий в комплект поставки Windows 7.

Выбор просматриваемого шрифта выполняется в раскрывающемся списке Шрифт. Если навести указатель мыши на один из символов, входящих в набор, и щелкнуть левой кнопкой, этот символ отображается в увеличенном виде. В строке состояния программы при этом появляется запись, указывающая код данного символа, а для некоторых стандартных шрифтовых наборов — также его описание и клавиатурная комбинация для ввода.

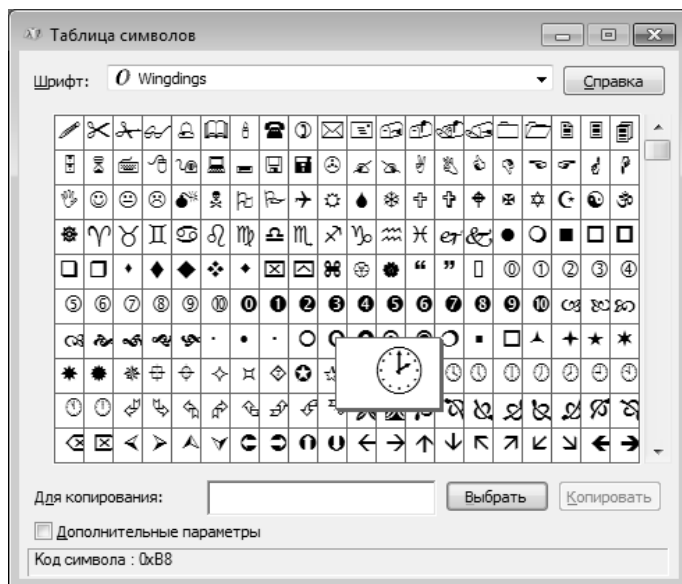


Рис. 7.14. Окно программы Таблица символов

Восстановление системы

Операционная система Windows имеет в своем составе средства, позволяющие восстановить ее безошибочную работу в случае повреждения каких-либо системных файлов. Механизм такого восстановления основан на создании так называемых *контрольных точек*, содержащих сведения о состоянии системы и копии важных системных файлов. При возникновении каких-либо неполадок можно воспользоваться существующей контрольной точкой и вернуть систему в прежнее работоспособное состояние.

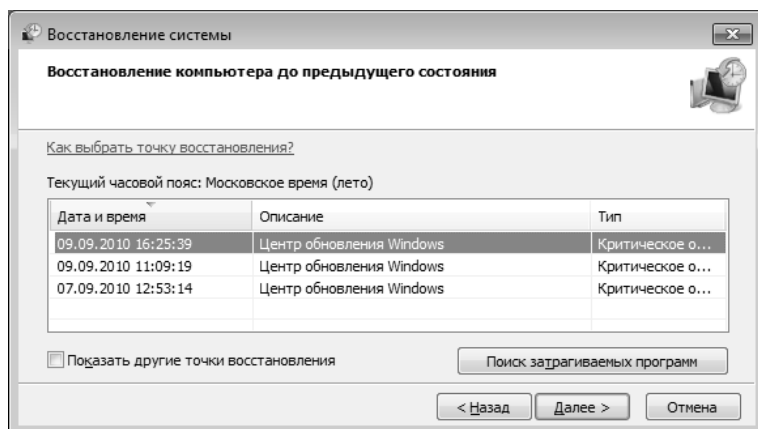


Рис. 7.15. Программа Восстановление системы позволяет вернуть состояние операционной системы к одной из ранних дат

Созданием контрольных точек и восстановлением системы руководит специальная программа — Восстановление системы. Операционная система автоматически создает контрольные точки, когда выполняются «опасные», на ее взгляд, операции. Кроме того, контрольную точку можно создать в любой момент по инициативе пользователя. Восстановление системы выполняется только по явной команде пользователя.

Недостатком механизма восстановления являются значительные затраты дискового пространства на хранение контрольных точек. Если аппаратная или программная конфигурация компьютера подвержена регулярным изменениям, то объем дискового пространства, используемый для хранения данных контрольных точек, может выйти за любые разумные пределы. В этом случае средство Восстановление системы можно отключить, отдавая себе отчет, что надежность работы компьютера в этом случае может снизиться. Снова активировать это средство можно в любой момент.

Наблюдение за функционированием компьютера и операционной системы

Операционная система Windows содержит специальный компонент для визуального и протокольного наблюдения за функционированием компьютерной системы. Этот компонент называется Монитор ресурсов (рис. 7.16).

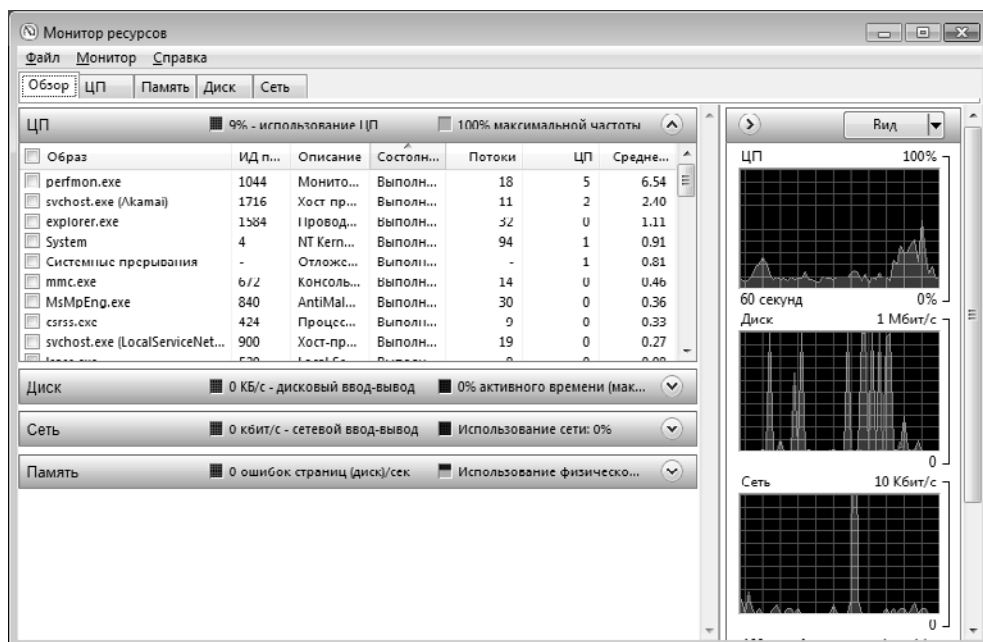


Рис. 7.16. Наблюдение за работой компьютера с помощью Монитора ресурсов

Программа позволяет контролировать загрузку процессора, распределение оперативной памяти, динамику обмена данными между накопителями и другие

параметры вычислительной системы. Результаты наблюдения отображаются на экране в виде графиков и заносятся в протокольный файл. Исследование компьютера с применением такого инструмента позволяет находить «узкие места» в его производительности, сравнивать между собой варианты настройки аппаратных и программных средств.

Средства командной строки

Ряд средств специфической настройки компьютера недоступны в системе Windows через Главное меню. Как правило, такие ограничения наложены на потенциально опасные и редко используемые средства. Предполагается, что этими средствами в случае необходимости воспользуется специалист по обслуживанию компьютера.

Запуск подобных средств осуществляется с командной строки. Команда и параметры вводятся либо в диалоговом окне Запуск программы (Пуск ► Все программы ► Стандартные ► Выполнить), либо в специальном окне Командная строка (Пуск ► Все программы ► Стандартные ► Командная строка). Вот некоторые из полезных программ, запускаемых таким образом.

- ❑ `regedit.exe` — программа для ручного редактирования Реестра Windows, служебной базы данных, содержащей сведения об аппаратно-программной конфигурации компьютера.
- ❑ `msconfig.exe` — программа, задающая настройки, управляющие процессом начальной загрузки операционной системы. Позволяет редактировать системные файлы конфигурации и на временной основе отключать и подключать драйверы и команды, используемые в ходе начальной загрузки системы.
- ❑ `Dxdiag.exe` — средство диагностики DirectX.

7.4. Стандартные средства мультимедиа

Мультимедиа — понятие комплексное. С одной стороны, оно подразумевает особый тип документов, а с другой — особый класс программного и аппаратного обеспечения.

Мультимедийные документы отличаются тем, что кроме традиционных текстовых и графических данных могут содержать звуковые и музыкальные объекты, анимированную графику (мультипликацию), а также видеофрагменты.

Мультимедийное программное обеспечение — это программные средства, предназначенные для создания и/или воспроизведения мультимедийных документов и объектов.

Мультимедийное аппаратное обеспечение — это оборудование, необходимое для создания, хранения и воспроизведения мультимедийного программного обеспечения. Исторически к нему относятся: звуковая карта, дисководы CD-ROM и звуковые колонки. Эту группу оборудования называют также базовым мультимедийным комплектом.

В последние годы класс аппаратных средств мультимедиа активно развивается. Так, в него вошли устройства для обработки телевизионных сигналов и воспроиз-

ведения телепрограмм (ТВ-тюнеры), аппаратные средства для обработки сжатой видеoinформации (MPEG-декодеры), дисководы для воспроизведения цифровых видеодисков (DVD), оборудование для записи компакт-дисков (CD-R/RW; DVD-R/RW) и многие другие устройства.

Управление громкостью

Базовым регулятором громкости компьютерной системы является программа Микшер громкости (рис. 7.17). Все средства управления громкостью, имеющиеся в иных программах, действуют только в пределах, первично заданных этой программой.

Управление программой выполняется через значок регулятора громкости на панели индикации.

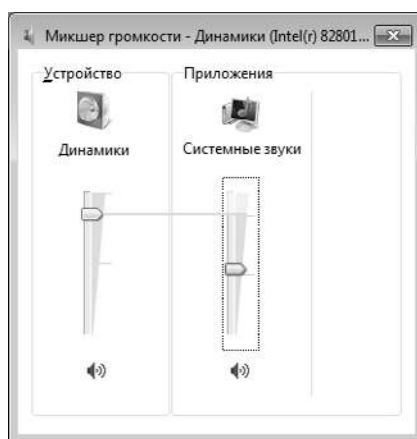


Рис. 7.17. Программа Микшер громкости позволяет отдельно регулировать уровень громкости звукового сигнала от разных приложений

Звукозапись

Программа Звукозапись предназначена для самостоятельного создания файлов звукозаписи. В качестве источника звука может использоваться микрофон или иное внешнее устройство, подключенное к звуковому адаптеру компьютерной системы. Программа запускается командой Пуск ► Все программы ► Стандартные ► Звукозапись.

Программа имеет минимальный набор графических элементов управления, позволяющих начать запись, остановить ее и снова продолжить. Записанный звук можно сохранить в файле формата WMA (Windows Media Audio). Программа позволяет создавать аудиоклипы небольших размеров, которые можно использовать в звуковых схемах оформления системных событий. Ее также используют в качестве OLE-сервера при необходимости вставить звуковой объект в текстовый документ.

Следует отметить, что в ранних операционных системах программа Звукозапись имела простейшие средства для управления качеством записи и ее редактирования. Начиная с операционной системы Windows Vista, средства управления качеством

звукозаписи вынесены отдельно. Теперь они находятся в диалоговом окне свойств устройства, подающего звуковой сигнал. Например, для настройки уровня громкости при записи с микрофона следует пользоваться элементами управления вкладки Пуск ► Панель управления ► Оборудование и звук ► Звук ► Запись ► Микрофон. В свою очередь, параметры качества записи выбирают в окне свойств микрофона, на вкладке Дополнительно, как показано на рис. 7.18.

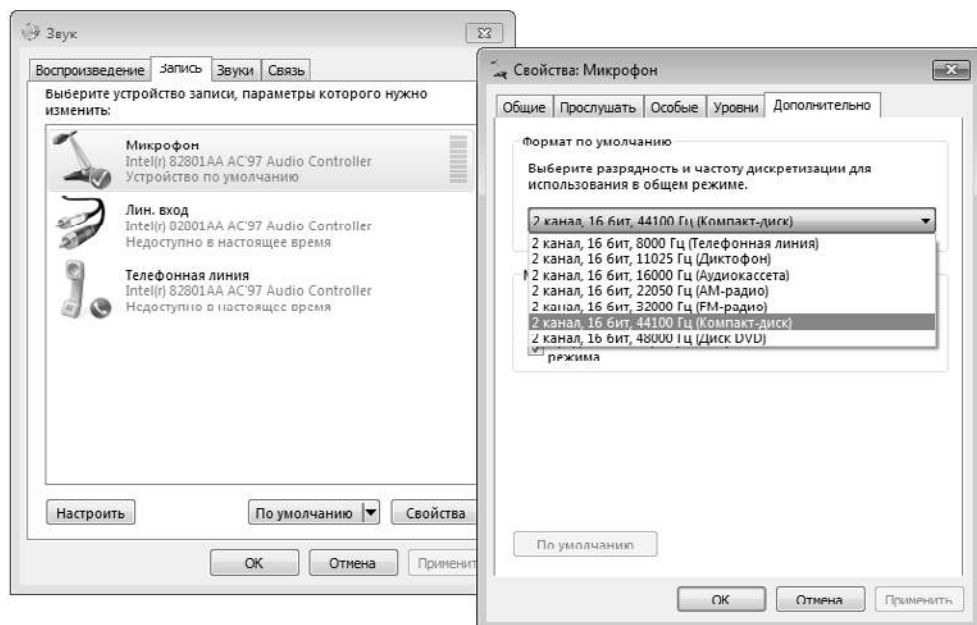


Рис. 7.18. Настройка параметров качества звукозаписи. Чем выше качество, тем больше размер полученного файла

Проигрыватель Windows Media

Проигрыватель Windows Media представляет собой универсальное средство для воспроизведения на компьютере всех видов видео- и аудиозаписей. В частности, с его помощью можно воспроизводить:

- ☐ музыкальные компакт-диски CD-ROM;
- ☐ видеодиски DVD-ROM;
- ☐ файлы аудио и видео;
- ☐ потоковые записи из Интернета.

В ранних версиях Windows эти функции возлагались на две отдельные программы: Лазерный проигрыватель (специально для музыкальных компакт-дисков) и Универсальный проигрыватель (для всех типов аудио- и видеофайлов). Возможность прослушивания потоковых звукозаписей Интернета появилась только в программе Проигрыватель Windows Media.

В основе Проигрывателя лежит локальная библиотека мультимедиа. Это база данных, в которой хранятся указатели на все мультимедийные ресурсы, имеющиеся на компьютере или когда-либо предъявленные Проигрывателю. То есть если звуко- или видеозапись уже воспроизводилась хотя бы однажды, она регистрируется в библиотеке и с ней можно обращаться средствами библиотеки (рис. 7.19).

Кроме того, в библиотеку автоматически заносятся сведения обо всех мультимедийных объектах, хранящихся в папках, которые Проигрыватель наблюдает автоматически. Списком этих папок можно управлять.



Рис. 7.19. Механизм управления Проигрывателем Windows Media основан на специальной базе данных — библиотеке мультимедиа

Программа имеет несколько режимов графического представления:

- ☐ режим библиотеки;
- ☐ режим компактного представления;
- ☐ режим обложки;
- ☐ режим текущего списка воспроизведения.

Режим библиотеки представления принят по умолчанию. Он используется для просмотра видеофильмов, а также для работы с библиотекой. На рис. 7.17 окно программы показано именно в режиме полного представления.

Режим компактного представления включается нажатием кнопки в правом нижнем углу окна программы. Этот режим наиболее удобен для воспроизведения звукозаписей.

Режим обложки и режим текущего списка воспроизведения предназначены для воспроизведения музыкальных и видеозаписей.

Комбинации клавиш CTRL+1, CTRL+2 и CTRL+3 активируют соответственно режимы библиотеки, обложки и текущего списка воспроизведения.

Экранные элементы управления Проигрывателя Windows Media соответствуют типовым органам управления бытовых электронных проигрывателей, магнитофонов, музыкальных центров. При воспроизведении музыкальных компакт-дисков Проигрыватель Windows Media способен загрузить из Интернета описание компакт-диска: название, имя автора или исполнителя, а также список дорожек. Предусмотрена также возможность копирования записей и сохранения их в сжатом формате .MP3.

В число дополнительных средств управления воспроизведением входят:

- ☐ средства управления объемным звучанием;
- ☐ графический эквалайзер;
- ☐ средства выбора зрительного образа;
- ☐ средства настройки видео.

Помимо файлов в мультимедийных форматах Проигрыватель Windows Media позволяет воспроизводить трансляции звукозаписей в потоковых форматах. Поточковая передача — это механизм доставки трансляций через сеть в непрерывном потоке. Обычные (файловые) записи сначала полностью загружают на компьютер, а потом уже воспроизводят. Поточковые же записи воспроизводят непосредственно в ходе их приема. При незначительных задержках поступления данных воспроизведение не прерывается, хотя некоторое ухудшение качества звучания возможно.

Для приема трансляции из Интернета следует указать адрес URL. Для этого дайте команду **Файл ► Открыть URL-адрес**. В диалоговом окне **Открытие URL-адреса** введите адрес ресурса и нажмите кнопку **ОК**. Воспроизведение потоковой записи начнется через несколько секунд после установки соединения. Это время необходимо на заполнение буфера данными.

Фотоальбом Windows Live

Фотоальбом Windows Live **поставляется вместе с бесплатным пакетом дополнительного программного обеспечения Windows Live. Он позволяет создавать виртуальные альбомы изображений, имеющихся на компьютере, и управлять ими вне зависимости от того, в каких именно папках эти изображения располагаются. Программа запускается командой Пуск ► Все программы ► Windows Live ► Фотоальбом Windows Live.**

Фотоальбом имеет три рабочих режима: обзора, просмотра и правки и слайд-фильма.

Режим обзора — основной режим работы Фотоальбома. Он предназначен для упорядочения коллекций фотоснимков (рис. 7.20). В этом режиме снимки представляются эскизами небольшого размера. Режим обзора принят по умолчанию: в этом режиме окно программы открывается при ее запуске.

Режим просмотра и правки предназначен для детального просмотра фотоснимков и их правки. Переход из режима обзора в режим просмотра и правки выполняется двойным щелчком на эскизе изображения, нажатием клавиши **ENTER** или с помощью командной кнопки Правка на панели инструментов.

Режим слайд-шоу предназначен для автоматизированной демонстрации серии предварительно отобранных фотоснимков в полноэкранном режиме. Предварительный отбор снимков выполняется путем группировки их в виртуальном альбоме. Режим слайд-фильма включается и выключается нажатием клавиши **F12**.

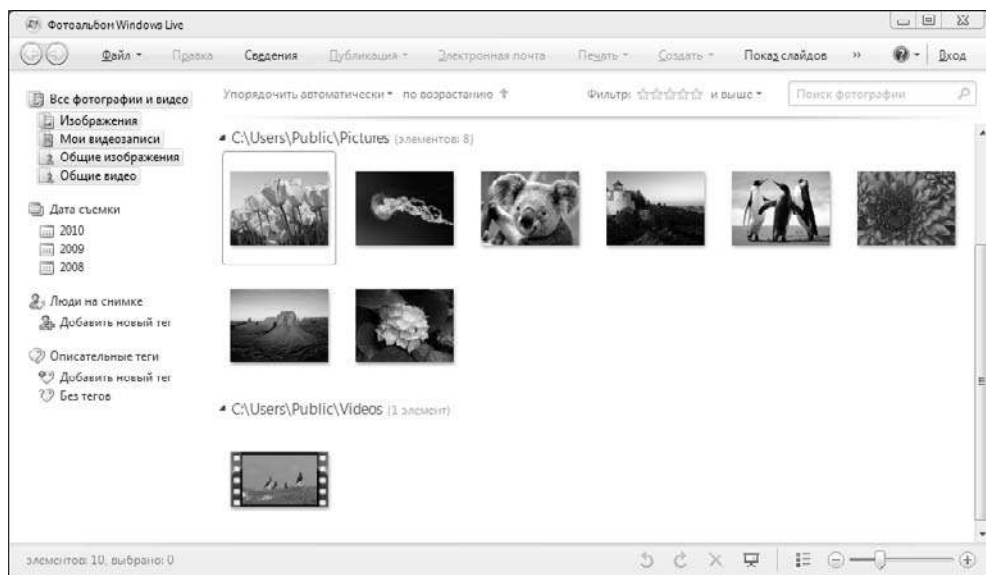


Рис. 7.20. Режим обзора — основной режим работы Фотоальбома. Он предназначен для обслуживания коллекций фотоснимков

Практическое занятие

Упражнение 7.1. Приемы работы с текстовым редактором Блокнот

Отработка приемов ввода и редактирования текстовых документов.

1. Запустите текстовый редактор Блокнот (Пуск ► Все программы ► Стандартные ► Блокнот).
2. Убедитесь, что включена русская раскладка клавиатуры. В противном случае щелкните на указателе языка на языковой панели и выберите в открывшемся меню пункт Русский. Если языковая панель закрыта, воспользуйтесь комбинацией клавиш, выбранной на данном компьютере.
3. Введите с клавиатуры слово Конденсатор (при вводе заглавной буквы удерживайте нажатой клавишу **SHIFT**) и нажмите клавишу **ENTER**.

4. Далее введите с клавиатуры термины Резистор, Катушка индуктивности, Выключатель, Амперметр и Вольтметр, нажимая после ввода каждого термина клавишу ENTER.
5. Расставьте в документе термины по алфавиту, выделяя строки и перемещая их через буфер обмена. Дважды щелкните на слове Амперметр и убедитесь, что оно при этом выделяется (в программе Блокнот этот способ служит для выделения отдельных слов) (рис. 7.21). Нажмите комбинацию клавиш SHIFT+ВПРАВО, чтобы включить в выделенный фрагмент невидимый символ конца строки — курсор при этом переместится в начало следующей строки.

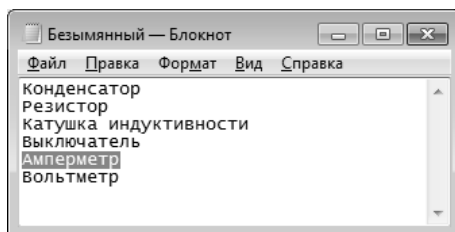


Рис. 7.21. Двойной щелчок выделяет слово целиком

6. Дайте команду Правка ► Вырезать, чтобы забрать выделенный фрагмент в буфер обмена. Убедитесь, что он действительно удаляется из документа.
7. Нажмите комбинацию клавиш CTRL+HOME, чтобы установить курсор в начало документа. Дайте команду Правка ► Вставить, чтобы вставить фрагмент из буфера обмена.
8. Установите указатель мыши на начало слова Вольтметр. Нажмите левую кнопку мыши и, не отпуская ее, выделите это слово методом протягивания.
9. Нажмите комбинацию клавиш CTRL+X, переместите текстовый курсор в начало второй строки текста и вставьте новый фрагмент из буфера обмена (CTRL+V).
10. Установите текстовый курсор в начало строки, содержащей слова Катушка индуктивности. Дважды нажмите комбинацию SHIFT+CTRL+ВПРАВО и убедитесь, что при каждом нажатии выделенный фрагмент расширяется, охватывая следующее слово. Нажмите комбинацию клавиш SHIFT+ ВПРАВО. Мы выделили нужный фрагмент при помощи клавиатурных команд.
11. Нажмите комбинацию клавиш SHIFT+DELETE, переместите текстовый курсор в начало третьей строки текста и вставьте новый фрагмент из буфера обмена с помощью комбинации клавиш SHIFT+INSERT.
12. Используя описанные приемы, завершите формирование списка введенных терминов в алфавитном порядке.
13. Сохраните созданный документ под именем list.txt.

Итог. Мы научились выполнять ввод и редактирование текстов в редакторе Блокнот. Мы освоили несколько приемов выделения и перемещения фрагментов текста через буфер обмена.

Упражнение 7.2. Приемы работы с графическим редактором Paint

В этом упражнении мы создадим условное обозначение вольтметра, принятое на электрических схемах.

1. Запустите графический редактор Paint (Пуск ► Все программы ► Стандартные ► Paint).
2. Убедитесь, что на палитре задан черный цвет в качестве основного и белый — в качестве фоновый.
3. Дайте команду Paint ► Свойства, в диалоговом окне Свойства изображения задайте ширину рисунка, равную 300 точек, и высоту — 200 точек. Щелкните на кнопке ОК.
4. Выберите инструмент Овал и в палитре настройки инструмента укажите вариант Без заливки.
5. Нажмите и удерживайте клавишу SHIFT. Методом протягивания нарисуйте окружность в центральной части области рисунка. Диаметр окружности должен составлять около половины высоты рисунка. Отпустите клавишу SHIFT.
6. Выберите инструмент Линия. В палитре настройки инструмента выберите вариант толщины линии (второй сверху).
7. Нажмите и удерживайте клавишу SHIFT. Методом протягивания нарисуйте небольшой горизонтальный отрезок прямой в стороне от окружности. Отпустите клавишу SHIFT.
8. Выберите инструмент Выделение. В палитре настройки инструмента выберите режим с прозрачным фоном.
9. Методом протягивания выделите прямоугольный фрагмент, охватывающий нарисованный отрезок прямой, но не затрагивающий окружность. Комбинацией клавиш CTRL+X поместите его в буфер обмена.
10. Вставьте отрезок прямой на рисунок комбинацией клавиш CTRL+V. Обратите внимание, что выделение при этом сохраняется.
11. Переместите выделенный фрагмент так, чтобы отрезок прямой примыкал к окружности слева. Обратите внимание на то, что фоновая часть фрагмента не перекрывает окружность.
12. Повторите операции, описанные в пп. 10–11, чтобы создать отрезок прямой, примыкающий к окружности справа.
13. Выберите инструмент Текст. Переключитесь на английскую раскладку клавиатуры.
14. Методом протягивания создайте область ввода текста внутри окружности. Введите символ «V». С помощью средств вкладки Текст задайте подходящий размер и начертание шрифта.
15. Методом перетаскивания за границу области ввода текста поместите букву «V» в центре окружности (рис. 7.22).
16. Щелкните вне области ввода текста, чтобы превратить текст в часть рисунка.

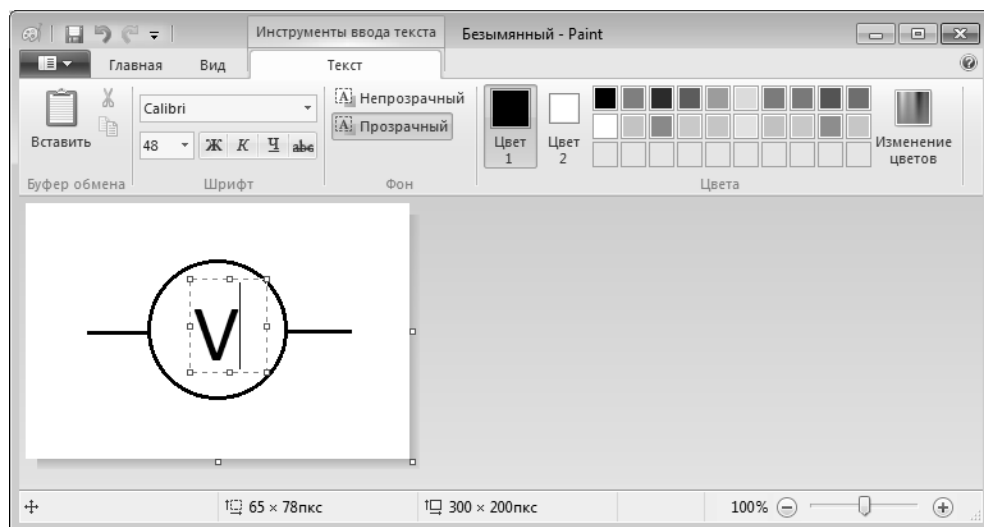


Рис. 7.22. Надпись на рисунке позиционируется методом перетаскивания

17. Сохраните созданное изображение под именем `scheme.png`.

Итог. Мы научились создавать простейшие примитивы (эллипс, линия), установили, как влияет регистровая клавиша `SHIFT` на работу инструментов рисования, научились вводить текстовые данные и компоновать рисунок из объектов.

Упражнение 7.3. Приемы форматирования в текстовом процессоре WordPad

Создать комплексный форматированный документ, содержащий, помимо текстовых, графические объекты.

В этом упражнении мы создадим иллюстрированный словарь терминов, введенных в файл `list.txt` в упражнении 7.1.

1. Запустите текстовый процессор WordPad (Пуск ► Все программы ► Стандартные ► WordPad).
2. Откройте текстовый файл `list.txt`.
3. Дайте команду WordPad ► Сохранить как, в списке Тип файла выберите пункт Файл RTF и сохраните файл под именем `dict.rtf`.
4. Выделите первое слово документа (Амперметр). На панели форматирования задайте шрифт Arial, размер шрифта — 14 пунктов, набор символов — Кириллический, выберите полужирное начертание.
5. Нажмите клавишу `END`, чтобы снять выделение, а затем — клавишу `ENTER`.
6. Введите краткое описание термина, указанного в предыдущей строке, например так: «прибор для измерения величины электрического тока». Размножьте введенный текст таким образом, чтобы образовался абзац размером 3–4 строки.

7. Выделите весь только что введенный абзац (можно использовать «тройной щелчок»). На панели форматирования задайте шрифт Times New Roman, размер шрифта — 12 пунктов, набор символов — Кириллический.
8. На линейке, расположенной ниже панели форматирования, перетащите маркер в виде квадратика на расстояние 1 см (по линейке) вправо. Убедитесь, что весь абзац теперь отображается с отступом от левого края.
9. Снимите выделение и установите курсор в начало первой строки того же самого абзаца. Нажмите клавишу TAB. Убедитесь, что табуляция в первой строке абзаца может использоваться для создания абзацного отступа.
10. Введите аналогичные краткие описания для последующих терминов создаваемого «словаря» и отформатируйте термины и описания так, как указано в пп. 4–9.
11. Установите курсор в конец описания термина Вольтметр и нажмите клавишу ENTER.
12. Дайте команду Вставка объекта. В диалоговом окне Вставка объекта включите переключатель Создать из файла.
13. Щелкните на кнопке Обзор, разыщите в файловой структуре ранее созданный документ scheme.png, щелкните на кнопке Открыть. Щелкните на кнопке ОК.
14. Убедитесь, что созданное изображение схематического обозначения вольтметра вставлено в документ в качестве иллюстрации.

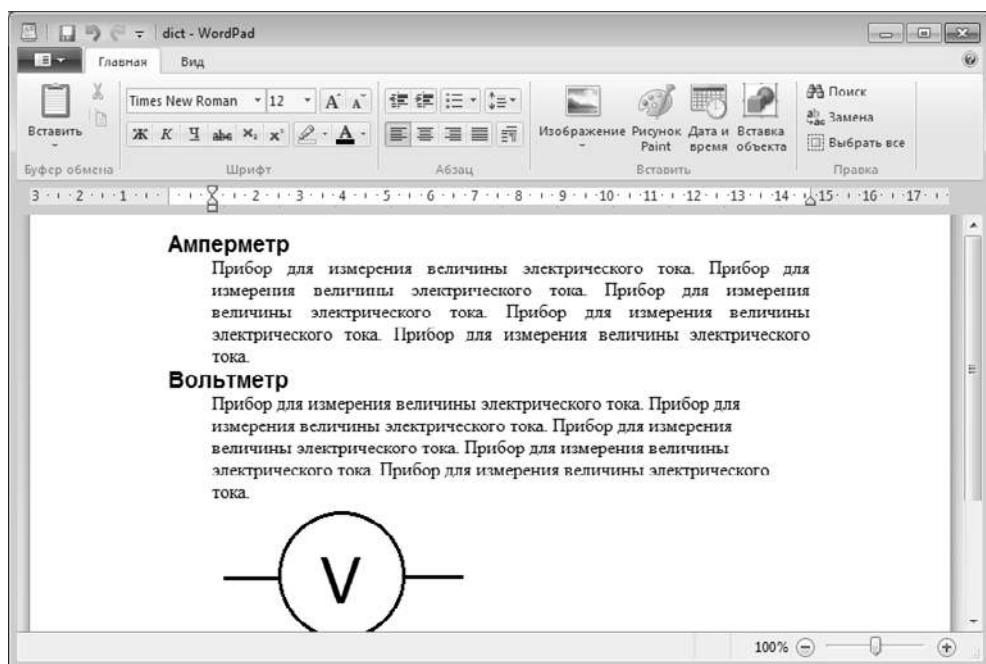


Рис. 7.23. Пример комплексного документа, содержащего встроенный объект

15. Измените масштаб отображения рисунка в документе путем перетаскивания маркеров изменения размера, расположенных на границах объекта.
16. Сохраните текущий документ dict.rtf.

Итог. Мы научились выполнять форматирование текста с помощью текстового процессора WordPad. В частности, мы научились по-разному оформлять заголовки и абзацы основного текста, а также встраивать графические объекты из внешнего источника.

Упражнение 7.4. Сопоставление приемов внедрения и связывания объектов

В предыдущем упражнении мы создали комбинированный документ dict.doc, содержащий внедренную иллюстрацию. В этом упражнении мы поместим тот же объект методом связывания.

1. Запустите текстовый процессор WordPad (Пуск ► Программы ► Стандартные ► WordPad). Откройте файл dict.rtf.
2. Удалите внедренный рисунок. Выделите его щелчком и нажмите клавишу DELETE.
3. Дайте команду Вставить объект. В диалоговом окне Вставка объекта установите переключатель Создать из файла.
4. Щелкните на кнопке Обзор и разыщите в файловой структуре документ scheme.png. Щелкните на кнопке Открыть — диалоговое окно Обзор закроется.
5. Установите флажок Связь, чтобы установить связь с рисунком (операция связывания). Щелкните на кнопке ОК.
6. Дайте команду WordPad ► Сохранить как и сохраните документ под именем dict1.rtf. Закройте программу WordPad.
7. Запустите программу Paint. Откройте файл scheme.png и измените его, например, закрасив внешнюю часть рисунка другим цветом с помощью инструмента Заливка. Сохраните рисунок scheme.png.
8. Запустите программу WordPad. Откройте документ dict.rtf. Проверьте, изменился ли его вид.
9. Закройте документ dict.rtf. Откройте документ dict1.rtf. Обратите внимание на вспомогательную операцию, выполняемую после загрузки документа. Проверьте, изменился ли вид этого документа. Чем вы объясните обнаруженные различия между документами?

Итог. Мы научились выполнять операцию связывания объектов с текстовым документом и исследовали различие между операциями внедрения и связывания.

Упражнение 7.5. Контроль загруженности процессора

Мониторинг производительности центрального процессора.

1. Запустите Монитор ресурсов (Пуск ► Все программы ► Стандартные ► Служебные ► Монитор ресурсов). Убедитесь, что в окне программы в графической

форме отображаются диаграммы загрузки центрального процессора (ЦП), жесткого диска (Диск), локальной сети (Сеть) и оперативной памяти (Память).

2. Подождите некоторое время, чтобы оценить загруженность процессора в отсутствие каких-либо активных действий (фактически она определяется необходимостью обслуживания самой программы Монитор ресурсов).
3. Откройте окно любой папки. Измените размер окна так, чтобы в нем помещалось 4–6 значков.
4. Наведите указатель мыши на строку заголовка открытого окна папки и в течение 10–20 секунд подвигайте окно по экрану, следя за показателями загрузки процессора в окне Монитора надежности и производительности. Запишите среднюю загрузку процессора во время этой операции.
5. Найдите на диске С: файл, размер которого более 10, но менее 50 Мб. Запишите среднюю нагрузку процессора в процессе поиска файла.
6. Создайте папку на Рабочем столе. Скопируйте в нее найденный файл. Запишите среднюю нагрузку процессора во время копирования файла.
7. Вновь наведите указатель мыши на строку заголовка окна открытой папки и в течение нескольких секунд подвигайте окно по экрану, следя за показателями в окне Монитора надежности и производительности. Запишите среднюю загрузку процессора во время этой операции.
8. Результаты эксперимента занесите в табл. 7.1.

Таблица 7.1. Измерение нагрузки процессора

	Дежурный режим	Перетаскивание окна	Поиск файла	Копирование файла
Средняя загрузка процессора, %				

Глава восьмая

Компьютерные сети, Интернет, компьютерная безопасность

8.1. Компьютерные сети

При физическом соединении двух и более компьютеров образуется *компьютерная сеть*. В общем случае, для создания компьютерной сети необходимо специальное аппаратное обеспечение (*сетевое оборудование*) и специальное программное обеспечение (*сетевые программные средства*). Для персональных моделей ПК сетевое оборудование обычно интегрировано в состав системной платы компьютера, а сетевые программные средства входят в состав операционной системы.

Назначение компьютерных сетей

Создание компьютерной сети имеет целью совместное использование ресурсов или совместный доступ к ним. В данном случае слово *ресурс* — очень емкое. В зависимости от назначения сети в него вкладывают различный смысл.

Сетевые ресурсы бывают трех типов: *аппаратные*, *программные* и *информационные*. Например, устройство печати (принтер) — это аппаратный ресурс. Когда все участники небольшой компьютерной сети пользуются общим принтером, они обобщают аппаратный ресурс. То же можно сказать и о сети, имеющей *файловый сервер*, на котором все участники сети хранят свои архивы и результаты работы.

Помимо аппаратных ресурсов, компьютерные сети позволяют совместно использовать *программные ресурсы*. Так, например, для выполнения очень сложных и продолжительных расчетов можно подключиться к удаленной большой ЭВМ и передать ей вычислительное задание, а по окончании расчетов получить их результат.

Данные, хранящиеся на удаленных компьютерах, образуют *информационный ресурс*. Роль этого ресурса сегодня видна наиболее ярко на примере Интернета, который многими воспринимается, прежде всего, как гигантская информационно-справочная система.

Наши примеры с делением ресурсов на аппаратные, программные и информационные достаточно условны. На самом деле, при работе в компьютерной сети любого типа одновременно происходит совместное использование всех типов ресурсов. Так, например, обращаясь в Интернет за справкой, мы безусловно используем чьи-то аппаратные средства, на которых работают чужие программы, обеспечивающие поставку затребованных нами данных.

Основные понятия компьютерных сетей

Основными задачами, решаемыми при создании компьютерных сетей, являются:

- ❑ обеспечение совместимости оборудования по электрическим и механическим характеристикам;

- обеспечение совместимости программного и информационного обеспечения по системе кодирования и формату данных.

Решение этих задач относится к области стандартизации и основано на модели взаимодействия открытых систем OSI (Open System Interconnections). Данная модель была разработана и утверждена Международным институтом стандартизации ISO (International Standards Organization) и называется моделью ISO/OSI.

Модель ISO/OSI

Согласно модели ISO/OSI, архитектуру компьютерных сетей рассматривают на нескольких уровнях. Обмен данными в такой модели происходит в три этапа: сначала в передающем устройстве данные перемещаются с верхнего уровня на нижний, затем они транспортируются к принимающему устройству и, наконец, в принимающем устройстве поднимаются с нижнего уровня на верхний (рис. 8.1).

Самый верхний уровень — прикладной. На этом уровне люди обмениваются информацией. Самый нижний уровень — физический, на нем устройства обмениваются друг с другом сигналами. Между верхним и нижним уровнями располагаются промежуточные уровни. На этих уровнях информация циркулирует в форме данных, причем на каждом уровне — свой формат данных.

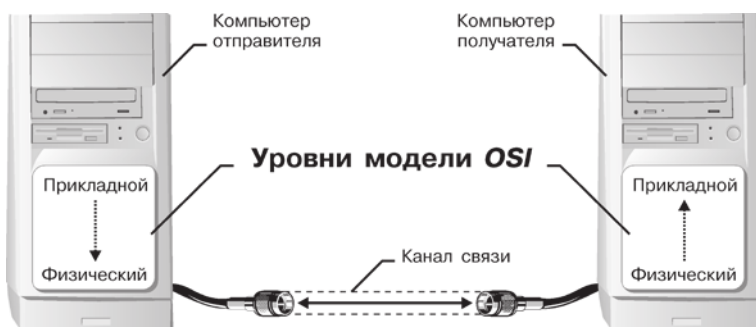


Рис. 8.1. Простейшая модель обмена данными в компьютерной сети

Протоколы связи

Для обеспечения необходимой совместимости на каждом из уровней компьютерной сети действуют свои стандарты, называемые протоколами. Аппаратные протоколы определяют характер взаимодействия устройств сети, а программные протоколы определяют условия взаимодействия программ и данных. Аппаратные устройства, обеспечивающие соблюдение протоколов, называют интерфейсами. А программные средства, играющие эту роль, называют программными протоколами или просто протоколами.

Так, например, если два компьютера соединены между собой, на физическом уровне протокол их взаимодействия определяют конкретные устройства физического порта и механические компоненты (разъемы, кабель и т. п.). На более высоких уровнях взаимодействие между компьютерами определяют программные средства, управляющие передачей данных через эти порты.

Локальные сети (LAN)

В соответствии с используемыми протоколами компьютерные сети принято разделять на локальные и глобальные. Характерная особенность локальной сети (LAN — Local Area Network) заключается в том, что все ее участники на нижних уровнях модели ISO/OSI используют единый комплект протоколов. Единство протоколов в элементах локальной сети обычно обеспечивается благодаря тому, что все они имеют общего собственника. Соответственно, сети этого вида имеют выраженную территориальную локализацию в пределах одного помещения, здания, группы компактно расположенных зданий.

Глобальные сети (WAN)

Глобальные сети (WAN — Wide Area Network), как правило, имеют неопределенные территориальные размеры, а их элементы могут принадлежать неопределенному кругу собственников. Глобальные сети связывают между собой как отдельные компьютеры, так и отдельные локальные сети, в том числе и использующие различные протоколы на нижних уровнях модели ISO/OSI.

Рабочая группа

Группы сотрудников, работающих над одним проектом в рамках локальной сети, называются рабочими группами. Одна локальная сеть может связывать несколько рабочих групп.

Политика сети

У участников рабочих групп могут быть разные права для доступа к общим ресурсам сети. Совокупность приемов разделения и ограничения прав участников компьютерной сети называется политикой сети.

Администрирование сети

Управление сетевыми политиками (их может быть несколько в одной сети) называется администрированием сети. Лицо, управляющее организацией работы участников локальной компьютерной сети, называется системным администратором.

Создание локальных сетей характерно для отдельных предприятий или их отдельных подразделений. Если же предприятие размещается на обширной территории, то его подразделения могут взаимодействовать через глобальные сети. В этом случае локальные сети подключаются к глобальной сети на основе договора присоединения, после чего присоединенные сети становятся частью глобальной сети.

Сетевой мост

Простейшее устройство для соединения между собой двух локальных сетей, использующих одинаковые протоколы, называется мостом. Мост может быть аппаратным (специализированный компьютер) или программным. Цель моста — не выпускать за пределы локальной сети данные, предназначенные для внутреннего потребления. Вне сети такие данные становятся «сетевым мусором», впуская занимающим каналы связи.

Сетевой шлюз

Для связи между собой нескольких локальных сетей, работающих по разным протоколам, служат специальные средства, называемые шлюзами. Шлюзы могут быть как аппаратными, так и программными. Например, это может быть специальный компьютер (шлюзовой сервер), а может быть и программа, установленная на обычный компьютер.

Сетевая безопасность

При подключении локальной сети предприятия к глобальной сети важную роль играет понятие сетевой безопасности. В частности, должен быть ограничен доступ в локальную сеть для посторонних лиц извне, а также ограничен выход за пределы локальной сети для сотрудников предприятия, не имеющих соответствующих прав.

Сетевой экран. Брандмауэр

Для обеспечения сетевой безопасности между локальной и глобальной сетью устанавливают так называемые сетевые экраны, или брандмауэры. Брандмауэром может быть устройство или программа, препятствующая несанкционированному перемещению данных между сетями.

8.2. Сетевые службы, основные понятия

Понятие виртуального соединения

Между участниками информационного обмена всегда существует физическая связь через материальные среды или средства. Однако в общественных отношениях рассматривать информационные связи через физические соединения не очень удобно. Например, корреспондентов из Москвы и Владивостока совершенно не интересуют технические детали сообщения между этими городами. Им совершенно все равно, будет ли почтовая бандероль доставлена самолетом или поездом: главное, чтобы она попала в руки адресата в заданный срок. Поэтому в общественных отношениях обычно вместо понятия физического соединения между участникам связи используют понятие виртуального соединения.

Рассмотрим простой пример взаимодействия двух корреспондентов с помощью обычной (не электронной) почты. Если они регулярно обмениваются письмами, то между ними действует виртуальное соединение. Оно было бы физическим, если бы каждый лично относил письмо партнеру и вручал в собственные руки, а не бросал его в почтовый ящик.

Сбором корреспонденции из общественных почтовых ящиков занимаются местные почтовые службы. Это следующий уровень модели связи. Далее, чтобы наше письмо достигло адресата в другом городе, должна существовать связь между нашей местной почтовой службой и его местной почтовой службой. Это еще один пример виртуальной связи, поскольку никакой физической связью эти службы не обладают — поступившую почтовую корреспонденцию они только сортируют и передают на уровень федеральной почтовой службы.

Федеральная почтовая служба в своей работе опирается на службы очередного уровня, например на почтово-багажную службу железнодорожного ведомства. И только рассмотрев работу этой службы, мы найдем, наконец, признаки физического соединения, например железнодорожный путь, связывающий два города.

Это очень простой пример, поскольку в реальности даже доставка обычного письма может затронуть гораздо большее количество служб. Но нам важно обратить внимание на то, что в нашем примере образовалось несколько виртуальных соединений между аналогичными службами, находящимися в пунктах отправки и приема. Не вступая в прямой контакт, эти службы взаимодействуют между собой. На каком-то уровне письма укладываются в мешки, мешки пломбируют, к ним прикладывают сопроводительные документы, которые где-то в другом городе изучаются и передаются на аналогичном уровне.

Модель взаимодействия открытых систем ISO/OSI

Выше мы сказали, что согласно рекомендациям Международного института стандартизации системы компьютерной связи рекомендуется рассматривать на семи разных уровнях. Теперь мы рассмотрим, как именно в этой модели происходит обмен данными между удаленными пользователями.

1. **Прикладной уровень.** На этом уровне пользователь работает с конкретным приложением. С помощью текстового процессора, графического редактора или иной прикладной программы он создает некое сообщение, содержащее нужную информацию.
2. **Уровень представления.** Этот уровень относится уже не к отдельной прикладной программе, а ко всей операционной системе компьютера в целом. На этом уровне информация, составляющая сообщение, представляется данными заданного формата. Происходит это одним из двух возможных способов: либо сохранением сообщения (в виде файла данных), либо внесением сообщения в базу данных (в виде записи базы данных).
3. **Сеансовый уровень.** Протоколы этого уровня проверяют права пользователя на подключение к сети, и если с правами все в порядке, передают документ к протоколам транспортного уровня. Если по условиям информационного обмена требуется обеспечение безопасности, на этом уровне происходит дополнительная операция: шифрование сообщения и приложение к нему сертификата электронной цифровой подписи (ЭЦП).
4. **Транспортный уровень.** Программы, обеспечивающие функционирование данного уровня, преобразуют числовые данные сообщения так, как это принято для используемой сети. При этом сообщение нарезается на небольшие пакеты стандартного размера. Каждый пакет получает заголовок с номером, который позволит восстановить сообщение на принимающем компьютере.
5. Протоколы **сетевого уровня** отвечают за маршрут движения данных в сети. Если на транспортном уровне сообщение было нарезано на пакеты, то на сетевом уровне каждый пакет должен получить адрес, по которому произойдет его доставка.

6. **Уровень соединения.** На этом уровне происходит переход от данных, представляющих сообщение, к сигналам, представляющим данные. Сигналы модулируются в соответствии с данными, полученными с сетевого уровня. В компьютере эти функции модуляции выполняет сетевой адаптер или модем.
7. **Физический уровень.** Реальный информационный обмен происходит на физическом уровне. Здесь нет ни документов, ни пакетов, ни даже байтов — только сигналы. Однако из этих сигналов принимающий компьютер способен выделить информационные биты и далее восстановить байты данных и воспроизвести сообщение. Восстановление информации происходит постепенно, при переходе с нижнего на верхний уровень модели OSI на принимающем компьютере.

Особенности виртуальных соединений

Разные уровни протоколов сервера и клиента не взаимодействуют друг с другом напрямую: они взаимодействуют через физический уровень. Постепенно переходя с верхнего уровня на нижний, данные непрерывно преобразуются, «обрастают» дополнительными данными, которые анализируются протоколами соответствующих уровней на сопредельной стороне. Это и создает эффект виртуального взаимодействия уровней между собой. Однако, несмотря на виртуальность, это все-таки соединения, через которые тоже проходят данные.

Это очень важный момент с точки зрения компьютерной безопасности. Одновременно с теми запросами на поставку данных, которые клиент направляет серверу, передается масса служебной информации, которая может быть как желательной, так и нежелательной. Например, обязательно передаются данные о текущем адресе клиента, о дате и времени запроса, о версии его операционной системы, о его правах доступа к запрашиваемым данным и пр. Передается и немало косвенной информации, например о том, по какому адресу он посылал предыдущий запрос. Известны случаи, когда даже передавались идентификационные коды процессоров компьютеров.

На использовании виртуальных соединений основаны такие позитивные свойства электронных систем связи, как возможность работать по одному физическому каналу сразу с несколькими серверами. Но на них же основаны и такие негативные средства, как «троянские программы». Троянская программа — это вредоносная программа, создающая во время сеансов связи виртуальные соединения для передачи данных о компьютере, на котором установлена. Среди этих данных может быть парольная информация, информация о содержании жесткого диска и т. п. В отличие от обычных компьютерных вирусов, троянские программы не производят разрушительных действий на компьютере и потому они лучше маскируются.

Сетевые службы

На виртуальных соединениях основаны все службы современного Интернета. Так, например, пересылка сообщения от сервера к клиенту может проходить через десятки различных компьютеров. Это совсем не означает, что на каждом компьютере сообщение должно пройти через все уровни — ему достаточно «подняться» до

сетевого уровня (определяющего адресацию) при приеме и вновь «опуститься» до физического уровня при передаче. В данном случае служба передачи сообщений основывается на виртуальном соединении сетевого уровня и соответствующих ему протоколах (рис. 8.2).

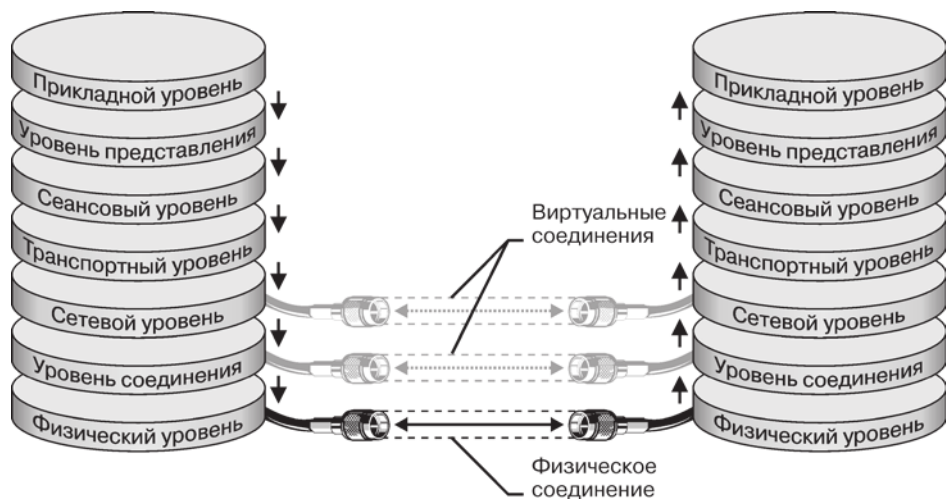


Рис. 8.2. Простейшая модель службы передачи сообщений

8.3. Интернет. Основные понятия

В дословном переводе на русский язык *интернет* — это *межсеть*, то есть в узком смысле слова интернет — это объединение сетей. Однако в 90-е годы XX века у этого слова появился и более широкий смысл: Всемирная компьютерная сеть. Интернет можно рассматривать в физическом смысле как сотни миллионов компьютеров, связанных друг с другом всевозможными линиями связи, однако такой «физический» взгляд на Интернет слишком узок. Лучше рассматривать Интернет как некое информационное пространство.

Интернет — это не совокупность прямых соединений между компьютерами. Так, например, если два компьютера, находящиеся на разных континентах, обмениваются данными в Интернете, это совсем не значит, что между ними действует одно прямое или виртуальное соединение. Данные, которые они посылают друг другу, разбиваются на пакеты, и даже в одном сеансе связи разные пакеты одного сообщения могут пройти разными маршрутами. Какими бы маршрутами ни двигались пакеты данных, они все равно достигнут пункта назначения и будут собраны вместе в цельный документ. При этом данные, отправленные позже, могут придти раньше, но это не помешает правильно собрать документ, поскольку каждый пакет имеет свою маркировку.

Таким образом, Интернет представляет собой как бы «пространство», внутри которого осуществляется непрерывная циркуляция данных. В этом смысле его можно сравнить с теле- и радиоэфиром, хотя есть очевидная разница хотя бы

в том, что в эфире никакая информация храниться не может, а в Интернете она перемещается между компьютерами, составляющими *узлы сети*, и какое-то время хранится на их жестких дисках.

Ранняя история Интернета

Ранние эксперименты по передаче и приему информации с помощью компьютеров начались еще в 50-х годах и имели лабораторный характер. В США решение о создании первой глобальной сети национального масштаба было принято в 1958 году. Оно стало реакцией на запуск в СССР первого искусственного спутника Земли.

Поводом для создания глобальной компьютерной сети стала разработка Пентагоном глобальной системы раннего оповещения о пусках ракет (*NORAD* — North American Aerospace Defense Command). Станции системы NORAD протянулись через север Канады от Аляски до Гренландии, а подземный командный центр расположился вблизи города Колорадо-Спрингс в недрах горы Шайенн. Центр управления был введен в действие в 1964 году, и, собственно, с этого времени можно говорить о работе первой централизованной глобальной компьютерной сети.

Курированием работы сети NORAD занималось Управление перспективных разработок министерства обороны США DARPA (Defense Advanced Research Project Agency). Ему было поручено преодолеть основной недостаток централизованной сети: недостаточную устойчивость, связанную с тем, что при выходе из строя какого-либо из узлов полностью выходил из строя и весь сектор, находившийся за ним, а при выходе из строя центра управления выходила из строя вся сеть. Во времена ядерного противостояния сверхдержав этот недостаток был критичным.

Основными направлениями исследований DARPA стали поиск новых протоколов обслуживания сети и новых принципов сетевой архитектуры. Первая вневедомственная национальная компьютерная сеть получила название ARPANET. Ее внедрение состоялось в 1969 году.

В 70-е годы развитие сети ARPANET происходило за счет подключения региональных сетей, воссоздающих общую архитектуру ARPANET на более низком уровне (в региональном или локальном масштабе). В этот период основной задачей ARPANET стала координация групп коллективов, работающих над едиными научно-техническими проектами, а основной функцией стал обмен электронной почтой и файлами с научной и проектной документацией. Одновременно продолжалась разработка новых сетевых протоколов, способных обеспечить живучесть глобальной сети даже в ядерном конфликте.

Внедрение протоколов TCP/IP

Всякий раз, когда мы говорим о вычислительной технике, нам надо иметь в виду принцип единства аппаратного и программного обеспечения. Пока глобальное расширение ARPANET происходило за счет механического подключения все новых и новых аппаратных средств (узлов и сетей), до Интернета в современном понимании этого слова было еще очень далеко.

Рождение Интернета в современном понимании произошло в 1983 году, когда была решена проблема устойчивости глобальной сети. Это произошло благодаря внедрению протоколов TCP/IP, лежащих в основе Всемирной сети по нынешний день. Решив эту задачу, управление DARPA прекратило свое участие в проекте и передало управление сетью Национальному научному фонду (NSF), который в США играет роль Академии наук. Так в 1983 году образовалась глобальная сеть NSFNET. В середине 80-х к ней начали активно подключаться академические и научные сети других стран, например академическая сеть Великобритании JANET (Joint Academic Network).

Образование системы доменных имен

Годы, когда глобальной сетью руководил Национальный научный фонд США, вошли в историю как эпоха решительной борьбы с попытками коммерциализации сети. Сеть финансировалась на правительственные средства. Национальный научный фонд распределял их между узлами и материально наказывал тех, кто пытался иметь от сети побочные доходы. В то же время развитие сети после внедрения протокола TCP/IP значительно ускорилось, NSF уже не успевал отслеживать деятельность каждого узла, а с подключением иностранных секторов его роль стала чисто символической.

Во второй половине 80-х годов произошло деление Всемирной сети на домены по принципу финансирования. Узлы домена GOV финансировались на средства правительства, а узлы доменов SCI и EDU — на средства научных учебных организаций. При этом особую роль приобрел домен COM (коммерческий). Он не финансировался никем, то есть его узлы должны были развиваться за счет собственных ресурсов. Национальные сети других государств стали рассматриваться как отдельные домены, например UK — домен Великобритании, а DE — домен Германии. Тогда и появилось понятие Интернета как саморазвивающейся децентрализованной иерархической структуры. Если во времена ARPANET и NSFNET сеть финансировалась сверху вниз, то теперь она финансируется от периферии, снизу вверх — от конечных пользователей к владельцам опорных сетей.

Основы функционирования Интернета

В основе функционирования Интернета лежит пара протоколов TCP/IP. В этой паре протокол TCP (Transmission Control Protocol) выполняет функции транспортного уровня: он отвечает за то, как происходит передача данных. В свою очередь, протокол IP (Internet Protocol) — это адресный протокол, выполняющий функции сетевого уровня. Он определяет, куда именно происходит передача.

Согласно протоколу TCP, отправляемые данные «нарезаются» на небольшие пакеты, после чего каждый пакет маркируется таким образом, чтобы в нем было все необходимое для правильной сборки документа на компьютере получателя.

Для понимания сути протокола TCP можно представить игру в шахматы по переписке, когда двое участников разыгрывают одновременно десяток партий. Каждый ход записывается на отдельной открытке с указанием номера партии и номера хода. В этом случае между двумя партнерами через один и тот же почтовый

канал работает как бы десяток виртуальных соединений (по одному соединению на каждую партию). Два компьютера, связанные между собой одним физическим соединением, могут точно так же поддерживать одновременно множество ТСП-соединений. Работая в Интернете, мы можем по одной физической линии связи принимать документы из Америки, Австралии и Европы практически одновременно. Пакеты каждого из документов поступают порознь, с разделением во времени, но прием всех документов происходит одновременно.

Теперь рассмотрим адресный протокол – IP. Он определяет способ записи адреса в Интернете для подключенных абонентских устройств. Такие устройства, обладающие уникальными адресами Интернета, называются *хостами* (как правило, это компьютеры, но не обязательно: хостами могут быть сетевые устройства, устройства мобильной связи, промышленные контроллеры и многие другие приборы). В свою очередь, уникальные интернет-адреса называются *IP-адресами*.

В настоящее время действуют две версии интернет-протокола: IPv4 и IPv6. Различаются они способами записи IP-адресов и, соответственно, методами их анализа. В версии IPv4 адрес хоста записывается 32 битами (четырьмя октетами по 8 бит в каждом, разделенными точками), например так: 195.38.46.11. Основным недостатком протокола IPv4 является ограниченность адресного пространства. С учетом того, что часть адресов отведена под служебные потребности, количество уникальных хостов не может превышать двух миллиардов. На заре Интернета эта величина казалась огромной, но для современного уровня развития информационного обмена этого совершенно недостаточно.

Адресное пространство протокола IPv6 в десятки раз шире, чем пространство IPv4, и в настоящее время может считаться неограниченным. Адрес хоста записывается 128 битами (восемь групп по четыре шестнадцатеричных числа, разделенные двоеточиями, например: 1234:5678:9abc:def0:0fed:cba9:8765:4321).

Службы Интернета

Когда говорят о работе в Интернете или об использовании Интернета, то на самом деле речь редко идет об Интернете в целом. Обычно имеется в виду одна из его многочисленных служб. В зависимости от конкретных целей и задач пользователи Сети используют те службы, которые им необходимы.

В простейшем понимании *сетевая служба* формируется парой программ, взаимодействующих между собой согласно определенным правилам, называемым протоколами. Одна из программ этой пары называется *сервером*, а вторая — *клиентом*. Соответственно, когда говорят о работе служб Интернета, речь идет о взаимодействии серверного оборудования и программного обеспечения и клиентского оборудования и программного обеспечения.

Разные службы имеют разные протоколы. Они называются *прикладными протоколами*. Их соблюдение обеспечивается и поддерживается работой специальных программ. Таким образом, чтобы воспользоваться какой-то из служб Интернета, необходимо установить на компьютере программу, способную работать по протоколу данной службы. Такие программы называют *клиентскими* или просто *клиентами*.

Для примера рассмотрим службу передачи файлов, которая позволяет пользователям эффективно обмениваться между собой файлами данных. Работа данной службы основана на специальном прикладном протоколе FTP (File Transfer Protocol). Соответственно, чтобы получить из Интернета файл, необходимо:

- ☐ иметь на компьютере программу, являющуюся FTP-клиентом;
- ☐ установить связь с компьютером, предоставляющим услуги FTP-сервера.

Аналогично, чтобы воспользоваться электронной почтой (E-Mail), необходимо соблюсти протоколы отправки и получения сообщений. Для этого надо иметь программу (*почтовый клиент*) и установить связь с *почтовым сервером*. Аналогично обстоит дело и с другими службами Интернета.

Служба World Wide Web

World Wide Web — это, безусловно, самая популярная служба современного Интернета. Ее нередко отождествляют с Интернетом в целом, хотя, конечно, это лишь одна из его многочисленных служб.

WWW. Служба World Wide Web образует и организует единое информационное пространство, состоящее из множества взаимосвязанных электронных документов, хранящихся на веб-серверах.

Веб-страница. Отдельные документы, составляющие пространство WWW, называются веб-страницами. Количество существующих веб-страниц измеряется миллиардами, но энергичный рост объема World Wide Web продолжается.

Веб-узел, сайт. Группы тематически объединенных веб-страниц, находящихся под общим администрированием, называют веб-узлами. Альтернативный термин — веб-сайт или просто сайт. Один физический веб-сервер может содержать достаточно много веб-узлов, каждому из которых, как правило, отводится отдельный каталог на жестком диске сервера.

От обычных текстовых документов веб-страницы отличаются тем, что они оформлены без привязки к конкретному носителю. Например, оформление документа, напечатанного на бумаге, привязано к параметрам печатного листа, который имеет определенную ширину, высоту и размер полей. Электронные веб-документы предназначены для просмотра на экране компьютера, причем заранее не известно, на каком именно. Не известны ни размер экрана, ни параметры его разрешения, ни даже операционная система, с которой работает компьютер клиента. Поэтому веб-документы не могут иметь «жесткого» форматирования. Оформление веб-страниц выполняется непосредственно во время их воспроизведения на компьютере клиента и происходит оно в соответствии с настройками программы, выполняющей просмотр.

Браузер. Программы для просмотра веб-страниц исторически называют браузерами. В период «неустойчивости» терминологии применялись также термины броузер или обозреватель, которые еще можно встретить в литературе. Во всех случаях речь идет о клиентской программе службы WWW, то есть о некотором средстве управления, обеспечивающем навигацию в пространстве WWW и просмотр веб-документов.

Разметка веб-страниц

Браузер выполняет отображение документа на экране, руководствуясь командами, которые автор документа внедрил в его текст (если автор применяет автоматические средства подготовки веб-документов, необходимые команды внедряются автоматически). Такие команды называются *тегами*. От обычного текста они отличаются тем, что заключены в угловые скобки. Большинство тегов используются парами: *открывающий* тег и *закрывающий*. Закрывающий тег начинается с символа «/».

<CENTER> Этот текст должен выравниваться по центру экрана </CENTER>

<LEFT> Этот текст выравнивается по левой границе экрана </LEFT>

<RIGHT> Этот текст выравнивается по правой границе экрана </RIGHT>

Сложные теги имеют кроме *ключевого слова* дополнительные *атрибуты* и *параметры*, детализирующие способ визуализации данных на экране. В частности, атрибуты и параметры позволяют выбрать гарнитуру шрифта для воспроизведения текста, задать размер символов, указать их цвет и начертание.

Язык разметки гипертекста, HTML

Правила записи тегов содержатся в спецификации особого *языка разметки*, близкого к языкам программирования. Он называется *языком разметки гипертекста* — *HTML* (HyperText Markup Language). Таким образом, веб-документ представляет собой обычный текстовый документ, размеченный тегами HTML. Такие документы также называют *HTML-документами* или *документами в формате HTML*.

При отображении HTML-документа на экране с помощью браузера теги не показываются, и мы видим только текст, составляющий документ. Однако оформление этого текста выполняется в соответствии с тем, какие теги имплантированы в текст документа.

Существуют специальные теги для внедрения графических и мультимедийных объектов (звук, музыка, видеоклипы). Встретив такой тег, браузер делает запрос к серверу на доставку файла, связанного с тегом, и воспроизводит его в соответствии с заданными атрибутами и параметрами тега — мы видим иллюстрацию или слышим звук.

Активные компоненты веб-страниц

В последние годы в веб-документах находят широкое применение так называемые *активные компоненты*. Это тоже объекты, но они содержат не только текстовые, графические и мультимедийные данные, но и программный код, то есть могут не просто отображаться на компьютере клиента, но и выполнять на нем работу по заложенной в них программе. Для того чтобы активные компоненты не могли выполнить на чужом компьютере разрушительные операции (что характерно для компьютерных вирусов), они исполняются только под контролем со стороны браузера. Браузер не должен допустить исполнения команд, несущих потенциальную угрозу: например, он пресекает попытки осуществить несанкционированные операции с жестким диском.

Гиперссылки

Возможность внедрения в текст графических и других объектов, реализуемая с помощью тегов, является одной из самых эффектных черт HTML с точки зрения оформления веб-страниц, но не самой важной с точки зрения самой идеи World Wide Web. Наиболее важной чертой веб-страниц, реализуемой с помощью тегов HTML, являются *гипертекстовые ссылки (гиперссылки)*. С любым фрагментом текста или, например, с рисунком с помощью тегов можно связать иной веб-документ, то есть установить *гиперссылку*. В этом случае при щелчке левой кнопкой мыши на тексте или рисунке, являющемся гиперссылкой, отправляется запрос на доставку и воспроизведение нового документа. Новый документ, в свою очередь, тоже может иметь гиперсвязи с другими документами.

Таким образом, гиперсвязи между электронными документами, хранящимися на серверах World Wide Web, образуют своеобразное гиперпространство документов, между которыми возможно перемещение. Документы, входящие в состав WWW, называют веб-ресурсами. Произвольное перемещение между веб-ресурсами называют *веб-серфингом*. Обычно веб-серфинг имеет целью ознакомительный просмотр веб-ресурсов. Целенаправленное перемещение между веб-документами называют *веб-навигацией*.

Унифицированный указатель ресурса, URL

Гипертекстовые связи между миллиардами документов, хранящихся на физических серверах Интернета, являются основой существования информационного пространства World Wide Web. А для того чтобы такие связи могли существовать, каждый документ в этом пространстве должен обладать своим уникальным адресом. Выше мы говорили, что каждый файл одного локального компьютера обладает *уникальным полным именем*, в которое входит собственное имя файла (включая расширение имени) и путь доступа к файлу, начиная от имени устройства, на котором он хранится. Теперь мы можем расширить представление об уникальном имени файла и развить его до Всемирной сети. Адрес любого файла во всемирном масштабе определяется *унифицированным указателем ресурса* — URL (Uniform Resource Locator).

Адрес URL состоит из трех частей: указания службы, предназначенной для доступа к данному ресурсу; указания доменного имени компьютера, на котором хранится данный ресурс, а также указания полного пути доступа к файлу на указанном компьютере.

1. Служба, которая осуществляет доступ к данному ресурсу, обозначается именем своего прикладного протокола с последующим знаком «двоеточие» и двумя знаками «косая черта». Так, например, для службы WWW прикладным является протокол HTTP (HyperText Transfer Protocol — протокол передачи гипертекста). Соответственно, первая часть URL-адреса для веб-ресурсов записывается так: `http://...`
2. Доменное имя компьютера (сервера), на котором хранится ресурс, следует читать справа налево. Наиболее правую позицию занимает доменное имя первого уровня, далее следует доменное имя второго уровня и т. д. Например, доменное

имя `www.piter-press.ru` принадлежит компьютеру, установленному в локальной сети издательского дома «Питер».

3. Запись полного пути доступа к файлу имеет несколько характерных особенностей. Во-первых, в качестве разделителя между папками используется символ «/» (косая черта). Для тех, кто привык, что в операционной системе Windows папки разделяются обратной косой чертой, это требование может показаться необычным, но не будем забывать, что большинство серверов Интернета используют операционные системы семейства UNIX, а не Windows. По этой же причине при записи пути доступа следует строго соблюдать регистр символов. В общем случае строчные и прописные символы считаются различными. Например, так выглядит URL-адрес файла `3dsMax.zip`, размещенного на веб-сервере издательского дома «Питер».

`http://www.piter-press.ru/books/Support/download/3dsMax.zip`

Именно в форме URL и связывают адрес ресурса с гипертекстовыми ссылками на веб-страницах. При щелчке на гиперссылке браузер посылает запрос для поиска и доставки ресурса, указанного в ссылке. Если по каким-то причинам ресурс не найден, выдается сообщение о том, что он недоступен (возможно, что сервер временно отключен или изменился адрес ресурса).

Служба имен доменов, DNS

Когда мы говорили о протоколах Интернета, то сказали, что адрес любого компьютера или любой локальной сети в Интернете может быть выражен в числовой форме согласно протоколу IP, например так:

`195.182.132.22`

В то же время мы заявили, что каждый компьютер имеет уникальное доменное имя, например такое:

`www.piter-press.ru`

Противоречия здесь нет, поскольку это просто две разные формы записи адреса одного и того же сетевого компьютера. Человеку неудобно работать с числовым представлением IP-адреса, зато доменное имя запоминается легко, особенно если учесть, что, как правило, это имя имеет содержание.

С другой стороны, автоматическая работа серверов сети организована с использованием четырехзначного числового адреса. Благодаря ему промежуточные серверы могут осуществлять передачу запросов и ответов в нужном направлении, не зная, где конкретно находятся отправитель и получатель.

Перевод доменных имен в связанные с ними IP-адреса выполняют серверы службы имен доменов DNS. Наш запрос на получение одной из страниц сервера `www.piter-press.ru` сначала обрабатывается сервером DNS, и далее он направляется по IP-адресу, а не по доменному имени.

Служба передачи файлов (FTP)

Прием и передача файлов составляют значительный процент от прочих Интернет-услуг. Необходимость в передаче файлов возникает, например, при приеме файлов

программ, при пересылке крупных документов (например, книг), а также при передаче архивных файлов, в которых запакованы большие объемы информации.

Служба FTP имеет свои серверы в мировой сети, на которых хранятся архивы данных. Со стороны клиента для работы с серверами FTP может быть установлено специальное программное обеспечение, хотя в большинстве случаев браузеры WWW обладают встроенными возможностями для работы и по протоколу FTP.

Протокол FTP работает одновременно с двумя ТСП-соединениями между сервером и клиентом. По одному соединению идет передача данных, а второе соединение используется как управляющее. Протокол FTP также предоставляет серверу средства для идентификации обратившегося клиента. Этим часто пользуются коммерческие серверы и серверы ограниченного доступа, поставляющие информацию только зарегистрированным клиентам, — они выдают запрос на ввод имени пользователя и связанного с ним пароля. Однако существуют и десятки тысяч FTP-серверов с *анонимным доступом* для всех желающих. В этом случае в качестве имени пользователя надо ввести слово: *anonymous*, а в качестве пароля задать адрес электронной почты. В большинстве случаев программы-клиенты FTP делают это автоматически.

ICQ

Эта служба — одна из нескольких существующих в Интернете служб для мгновенного обмена сообщениями. Если два человека подключены к Интернету одновременно, то, в принципе, им почти ничто не мешает общаться друг с другом напрямую. Единственная проблема — знание сетевого IP-адреса человека, подключенного в данный момент к Интернету. Большинство пользователей не имеет постоянного IP-адреса — такой адрес выдается им на временной основе в момент установки соединения. Название службы *ICQ* является акронимом выражения *I seek you* (Я тебя ищу). Для пользования этой службой надо зарегистрироваться на ее центральном сервере (<http://www.icq.com>) и получить персональный идентификационный номер *UIN* (Universal Internet Number). Данный номер можно сообщить партнерам по контактам. Зная номер *UIN* партнера, но не зная его текущий IP-адрес, можно через центральный сервер службы отправить ему сообщение с предложением установить соединение.

Как было указано выше, каждый компьютер, подключенный к Интернету, должен иметь четырехзначный IP-адрес. Этот адрес может быть *постоянным* или *динамически* временным. Те компьютеры, которые включены в Интернет на постоянной основе, имеют постоянные IP-адреса. Большинство же пользователей подключаются к Интернету лишь на время сеанса. Им выдается динамический IP-адрес, действующий только в течение данного сеанса. Этот адрес выдает тот сервер, через который происходит подключение. В разных сеансах динамический IP-адрес может быть различным, причем заранее не известно каким.

При каждом подключении к Интернету программа ICQ, установленная на нашем компьютере, определяет текущий IP-адрес и сообщает его центральной службе, которая, в свою очередь, оповещает наших партнеров по контактам. Далее наши партнеры (если они тоже являются клиентами данной службы) могут установить

с нами прямую связь. Программа предоставляет возможность выбора режима связи («готов к контакту»; «прошу не беспокоить, но готов принять срочное сообщение»; «закрыт для контакта» и т. п.). После установления контакта связь происходит в режиме, аналогичном сервису IRC.

8.4. Технологии проводного подключения к Интернету

Подключение к Интернету организуется по принципу присоединения. То есть, чтобы подключить свой компьютер к Интернету, следует на договорной основе подключить его к другому компьютеру или локальной сети, ранее подключенной к Интернету. Для этого существует несколько альтернативных технологий. Различаются они в первую очередь доступностью физической линии связи, производительностью канала связи и стоимостью его эксплуатации.

Помимо физического подключения к одному из узлов Всемирной сети для работы в Интернете необходимо:

- ☐ получить IP-адрес на постоянной или временной основе;
- ☐ установить и настроить программное обеспечение — программы-клиенты тех служб Интернета, услугами которых предполагается пользоваться.

Организации, предоставляющие возможность подключения к своему узлу и передающие в аренду свои IP-адреса, называются поставщиками услуг Интернета (используется также термин сервис-провайдер или просто провайдер). Мы будем далее использовать понятие оператор связи.

Физическое подключение может быть выделенным или коммутируемым. Для выделенного соединения необходимо, как правило, проложить новую или арендовать готовую физическую линию связи (кабельную, оптоволоконную, радиоканал, спутниковый канал и т. п.). От типа линии связи зависит ее пропускная способность (измеряется в единицах бит в секунду). В настоящее время пропускная способность мощных линий связи (оптоволоконных и спутниковых) составляет сотни мегабит в секунду (Мбит/с).

Коммутируемое телефонное подключение

В противоположность выделенному соединению, коммутируемое соединение — временное. Оно не требует специальной линии связи и может быть осуществлено, например, по телефонной линии. Коммутацию (подключение) выполняет автоматическая телефонная станция (АТС) по сигналам, выданным в момент набора телефонного номера.

Для телефонных линий связи характерна низкая пропускная способность. В зависимости от того, какое оборудование использовано на станциях АТС по пути следования сигнала, различают *аналоговые* и *цифровые* телефонные линии. Основную часть телефонных линий во многих городах России составляют устаревшие аналоговые линии. Их предельная пропускная способность не превосходит 50 Кбит/с (примерно две страницы текста в секунду или одна-две фотографии стандартного размера в минуту). Пропускная способность цифровых телефонных линий составляет 60–120 Кбит/с, то есть в 2–4 раза выше.

Телефонные линии связи никогда не предназначались для передачи цифровых сигналов — их характеристики подходят только для передачи голоса, причем в достаточно узком диапазоне частот — 300–3000 Гц. Поэтому для передачи цифровой информации несущие сигналы звуковой частоты *модулируют* по амплитуде, фазе и частоте. Такое преобразование выполняет специальное устройство — *модем* (название образовано от слов *модулятор* и *демодулятор*).

Основные достоинства коммутируемого соединения — экономичность и организационная доступность. Его основные недостатки:

- ☐ низкая производительность;
- ☐ недостаточная надежность;
- ☐ занятость телефонной линии во время сеанса связи с Интернетом.

В настоящее время коммутируемое соединение считается устаревшей технологией, но им продолжают пользоваться там, где другие технологии экономически нецелесообразны или технически недоступны.

Выделенный канал на телефонной линии

Основные недостатки коммутируемого соединения успешно преодолевает семейство технологий xDSL, превращающих обычную телефонную линию в аналог выделенной линии. В этом семействе наибольшую популярность приобрела технология ADSL (Asymmetric Digital Subscriber Line — асимметричная цифровая абонентская линия).

В основе этой технологии лежит тот факт, что реальные возможности медного телефонного провода гораздо выше тех скромных характеристик, которые обеспечивает телефонный модем. Фактически, если бы удалось исключить движение сигналов через коммутационные устройства АТС и напрямую направить их в интернет-каналы, то данные можно было бы передавать со скоростями в сто и более раз выше, чем позволяет коммутируемое телефонное соединение. Для этого необходимо, чтобы оператор связи установил на ближайшей к пользователю АТС специальное оборудование, которое разделяет приходящие сигналы на два потока: речевой (телефонный) и цифровой (компьютерный). Речевые данные направляются в телефонную сеть, а компьютерные — в магистральные каналы Интернета.

Соединение ADSL — несимметричное в том смысле, что скорость входящего потока в несколько раз выше, чем скорость исходящего потока. Это связано с экономическими соображениями и имеет целью снизить издержки конечных пользователей на оборудование. Стандартная величина входящего потока — до 8 Мбит/с, а исходящего — до 1 Мбит/с. Продолжением технологии ADSL стали более совершенные технологии ADSL2 и ADSL2+, позволяющие увеличить скорость исходящего потока до 1,5 Мбит/с, а входящего — до 24 Мбит/с.

К основным достоинствам ADSL-соединения относятся:

- ☐ создание выделенного канала связи без необходимости сооружения выделенной линии связи;

- ❑ высокая производительность и широкая функциональность, например через канал ADSL возможна трансляция телевизионных передач, а через канал ADSL2 — трансляция передач телевидения высокой четкости (HDTV);
- ❑ возможность одновременного использования телефона и Интернета.

В свою очередь, основным недостатком ADSL-соединения является недостаточная надежность связи и высокая степень зависимости качества услуги от технического состояния телефонной линии абонента.

Сетевое подключение

Наиболее распространенной технологией построения локальных вычислительных сетей является технология *Ethernet*. Согласно этой технологии компьютер абонента напрямую подключается к локальной сети оператора связи и становится ее частью. Для подключения он должен иметь стандартный адаптер локальной сети или материнскую плату со встроенным адаптером. В качестве среды, несущей сигналы, используется коаксиальный кабель, волоконно-оптический кабель или провод «витая пара».

Стандартная скорость передачи данных при сетевом подключении составляет:

- ❑ 10 Мбит/с (Ethernet);
- ❑ 100 Мбит/с (Fast Ethernet);
- ❑ 1000 Мбит/с (Gigabit Ethernet);
- ❑ 10 Гбит/с (10 Gbit Ethernet)

Обычно оператор связи заранее устанавливает свои маршрутизаторы в местах предполагаемого подключения абонентов: в зданиях, сооружениях, подъездах жилых домов. Для подключения абонента от его компьютера до ближайшего маршрутизатора прокладывается проводная или беспроводная линия связи.

Основными достоинствами подключения по технологии Ethernet являются: высокая надежность, производительность и функциональность соединения. Основным недостатком — высокая стоимость сооружения линии, если вблизи потребителя нет заранее развернутой сетевой инфраструктуры.

Кабельное подключение

Этот вид подключения основан на использовании кабельных сетей телевизионного вещания. Соответственно, он находит основное применение в населенных пунктах с развитой инфраструктурой средств связи.

Компьютер сопрягается с кабельной сетью через специальный адаптер, который называют *кабельным модемом*. Со стороны сети к нему подходит телевизионный кабель, а со стороны ПК — кабель локальной сети Ethernet. Существуют также кабельные модемы, связывающиеся с компьютером по беспроводной технологии Wi-Fi.

Свойства коаксиального кабеля позволяют передавать данные с очень высокими скоростями — до нескольких гигабит в секунду, но реальная производительность соединения ограничена возможностями кабельного модема. Обычная величина — десятки мегабит в секунду.

Основное достоинство кабельного соединения: высокая производительность и надежность. Основной недостаток — потребность в специальном оборудовании и наличии развернутой инфраструктуры связи. Обычно этот вид подключения осуществляется как дополнительная услуга для абонентов услуг кабельного телевидения.

8.5. Технологии беспроводного подключения к Интернету

В последние годы технологии беспроводного доступа к Интернету развиваются ускоренными темпами. Их основное преимущество: простота подключения к действующей инфраструктуре (не надо прокладывать физическую линию связи к абоненту). Их основной недостаток — необходимость специальных мер защиты соединения от подключения посторонних лиц.

Соединение GPRS

GPRS — это технология пакетной передачи данных в сетях мобильной телефонии GSM. При установлении GPRS-соединения оператор мобильной связи выделяет радиотелефону абонента постоянный или временный IP-адрес.

Теоретическая производительность GPRS-соединения составляет примерно 170 Кбит/с, но на практике производительность зависит как от текущей загруженности сети, так и от политики работы оператора. В большинстве случаев операторы мобильной связи отдают приоритет речевому трафику перед трафиком данных. Поэтому эффективная производительность соединения обычно составляет 40–50 Кбит/с.

Основными недостатками GPRS-технологии являются:

- ❑ малая пропускная способность соединения;
- ❑ недостаточная надежность;
- ❑ недостаточная доступность (связь действует на расстоянии не более 2–3 км от базовой станции сети).

Однако доступность является также и основным достоинством технологии. GPRS-соединение с Интернетом можно использовать там, где нет никакой стационарной инфраструктуры связи: в сельской местности, в местах летнего отдыха, в районах новостроек и даже в движущемся транспорте.

Технологии GSM-связи постоянно развиваются, поэтому сегодня абонентам, желающим использовать для доступа в Интернет средства сотовых сетей, может быть доступен не только GPRS, но и более современные, скоростные и надежные технологии. В частности, речь идет о сервисах, предоставляемых так называемыми 3G-сетями, GSM-сетями третьего поколения. Скорость передачи данных в таких сетях может достигать нескольких мегабит в секунду.

Соединение Wi-Fi

Технология Wi-Fi (Wireless Fidelity) основана на семействе стандартов беспроводной передачи данных по радиоканалу. Рабочая частота радиоканала Wi-Fi 2,4 ГГц. Радиус действия соединения 30–40 м в помещении и до 100 м на открытой местности. Производительность соединения определяется конкретным стандартом:

- ❑ до 11 Мбит/с (стандарт 802.11b);
- ❑ до 54 Мбит/с (стандарт 802.11g);
- ❑ до 300 Мбит/с (стандарт 802.11n).

Главное достоинство технологии Wi-Fi — доступность. Точки доступа Wi-Fi, к которым возможно простое подключение, широко доступны в местах массового пребывания людей: в гостиницах, библиотеках, кафе, барах, на вокзалах и станциях. Важным преимуществом для малых офисов и личных домохозяйств является также экономичность и удобство подключения: нет необходимости прокладывать проводную линию связи, выполнять ремонтно-строительные работы, нарушать интерьер и дизайн помещений.

Основными недостатками беспроводного соединения Wi-Fi являются: сравнительно невысокий радиус действия соединения и необходимость обеспечения сетевой безопасности для защиты соединения от подключения посторонних лиц.

Соединение WiMAX

Телекоммуникационная технология WiMAX (Wireless Interoperability for Microwave Access) предназначена для беспроводного подключения к Интернету широкого спектра устройств от стационарных до мобильных. Она призвана парировать основной недостаток технологии Wi-Fi: ограниченный радиус действия соединения. Технология WiMAX особенно эффективна, когда решается не частная задача подключения отдельного абонента к Интернету, а необходимо оперативно развернуть опорную сеть для создания инфраструктуры связи в целом регионе.

Узлы опорной сети WiMAX располагаются на вышках или на крышах сооружений таким образом, чтобы обеспечивалось условие прямой видимости между соседними станциями. При этом расстояние между ними может достигать десятков километров, а скорость обмена данными — 120 Мбит/с.

Абонентские станции сетей WiMAX могут иметь как стационарное, так и мобильное исполнение. Для связи с базовой станцией наличие прямой видимости не является обязательным условием. Радиоканал работает на частоте от 1,5 до 11 ГГц. Радиус зоны уверенного приема составляет от 3 до 10 км в зависимости от конструктивного исполнения абонентской станции, условий ее размещения, а также окружающего ландшафта. Скорость обмена данными достигает 45 Мбит/с.

8.6. Информационная безопасность в Интернете

Работая в Интернете, следует понимать, что насколько ресурсы Всемирной сети открыты ее абонентам, настолько же и ресурсы их систем при определенных условиях могут быть доступны тем, кто располагает средствами, необходимыми для доступа к ним. Поэтому работа в Интернете должна сопровождаться мерами, направленными на обеспечение информационной безопасности. При этом следует иметь в виду два аспекта проблемы:

- ❑ предотвращение несанкционированного удаленного администрирования компьютерной системы;
- ❑ обеспечение целостности электронного документооборота.

Защита от удаленного администрирования

Под несанкционированным удаленным администрированием понимается получение незаконного доступа к охраняемой информации программными средствами. В большинстве случаев это достигается размещением на атакуемом компьютере вредоносного кода. Доставка такого кода осуществляется либо скрытным встраиванием его в системное или прикладное программное обеспечение (*программная закладка*), либо автоматизированным распространением с помощью вредоносных программных средств (компьютерных вирусов), использующих несовершенство программного обеспечения.

Защита от программных закладок осуществляется организационно-техническими средствами. Прежде всего следует исключить эксплуатацию программных средств, полученных из непроверенных источников. Эта задача решается запретом установки и запуска программного обеспечения для всех пользователей ПК, не являющихся администраторами. Современные операционные системы такую возможность предоставляют.

Защита от компьютерных вирусов

Компьютерный вирус — это программный или сценарный код, встроенный в программу, сообщение, документ или в определенную область носителя данных и предназначенный для выполнения несанкционированных действий на несущем компьютере. Основными типами компьютерных вирусов являются: программные вирусы, загрузочные вирусы, макровирусы, почтовые вирусы.

Профилактика программных вирусов

Программные вирусы — это блоки вредоносного программного кода, целенаправленно внедренные в код обычных программных средств. При запуске программы, несущей вирус, происходит запуск имплантированного в нее вирусного кода. Работа этого кода вызывает скрытые от пользователя изменения в файловой системе жестких дисков и/или в содержании других программ. Так, например, вирусный код может воспроизводить себя в теле других программ — этот процесс называется размножением. По прошествии определенного времени, создав достаточное количество копий, программный вирус может перейти к разрушительным действиям — нарушению работы программ и операционной системы, удалению информации, хранящейся на жестком диске. Этот процесс называется вирусной атакой.

Самые разрушительные вирусы могут инициировать форматирование жестких дисков. Поскольку форматирование диска — достаточно продолжительный процесс, который не может пройти незамеченным со стороны пользователя, во многих случаях программные вирусы ограничиваются уничтожением данных только в системных секторах носителя, что эквивалентно потере таблиц файловой структуры. В этом случае данные на жестком диске остаются нетронутыми, но воспользоваться ими нельзя без применения специальных средств, поскольку не известно, какие сектора диска каким файлам принадлежат. Теоретически восстановить данные в этом случае можно, но трудоемкость этих работ достаточно высока.

Программные вирусы поступают на компьютер при запуске непроверенных программ, полученных на внешнем носителе или принятых из Интернета. Особое внимание следует обратить на слова «при запуске». При обычном копировании

данных заражение компьютера не происходит. В связи с этим все данные, принятые из Интернета, должны проходить обязательную проверку на безопасность, а если получены незатребованные данные из незнакомого источника, их следует уничтожить, не рассматривая.

Профилактика загрузочных вирусов

Загрузочные вирусы отличаются от программных методом распространения. Они поражают не файлы программ, а определенные системные области носителей данных. Кроме того, на включенном компьютере они могут временно располагаться в оперативной памяти.

Обычно заражение происходит при попытке загрузки компьютера с носителя, системная область которого содержит загрузочный вирус. При этом сначала происходит проникновение вируса в оперативную память, а затем в загрузочный сектор жесткого диска, после чего компьютер становится источником распространения загрузочного вируса.

Основная мера профилактики: обязательная проверка носителей данных перед применением. Проверка выполняется посредством сканирования с помощью антивирусного программного обеспечения.

Профилактика макровирусов

Эта особая разновидность вирусов поражает документы, выполненные в некоторых прикладных программах, имеющих средства для исполнения так называемых *макрокоманд*. В частности, к таким документам относятся документы текстового процессора Microsoft Word. Заражение происходит при открытии файла документа в окне программы, если в ней не отключена возможность исполнения макрокоманд. Как и для других типов вирусов, результат атаки может быть как относительно безобидным, так и разрушительным.

Основные меры профилактики: контроль за настройками текстового процессора, отказ от использования стандартных шаблонов и обязательное сканирование поступающих документов с помощью антивирусной программы.

Профилактика почтовых вирусов

В настоящее время электронная почта является наиболее популярным средством для распространения компьютерных вирусов. Здесь важно обратить внимание на то, что заражение вирусами этой категории происходит не в результате поступления почтового сообщения, а наступает в результате некорректных действий пользователя при просмотре вредоносного сообщения и связано с наличием ошибок и уязвимостей в почтовых программах и операционных системах.

Соответственно, профилактические мероприятия включают:

- ☐ регулярное обновление почтовой программы и операционной системы;
- ☐ корректное обращение со всеми почтовыми вложениями, прикрепленными к основному сообщению (вложения, полученные из неизвестных источников, следует удалять, не открывая);
- ☐ адекватную настройку почтовой программы, препятствующую автоматическому воспроизведению сообщений и вложений.

Организация резервного копирования данных

Важным средством защиты информации является резервное копирование наиболее ценных данных. Наличие резервной копии позволяет восстановить данные в случае их утраты. Восстановление компьютерной системы происходит в следующем порядке.

1. Жесткие диски переформатируют и подготавливают к новой эксплуатации.
2. На «чистый» отформатированный диск устанавливают операционную систему с дистрибутивного компакт-диска.
3. Далее под управлением операционной системы устанавливают все необходимое программное обеспечение, которое тоже берут с дистрибутивных носителей.
4. Восстановление компьютера завершается восстановлением данных, взятых с резервных носителей.

При резервировании данных следует также иметь в виду, что следует сохранить все регистрационные и парольные данные для доступа к сетевым службам Интернета. Их не следует хранить на компьютере. Обычное место их хранения — служебный дневник в сейфе руководителя подразделения.

Создавая план мероприятий по резервному копированию информации, необходимо учитывать, что резервные копии должны храниться отдельно от компьютера. То есть, например, резервирование информации на отдельном жестком диске того же компьютера лишь создает иллюзию безопасности. Относительно новым и достаточно удобным средством хранения данных являются веб-папки, размещенные на удаленных серверах в Интернете. Существуют службы, бесплатно предоставляющие пространство для хранения данных пользователя.

Резервные копии конфиденциальных данных сохраняют на внешних носителях, которые хранят в сейфах, желательно в отдельных помещениях. При разработке организационного плана резервного копирования учитывают необходимость создания не менее двух резервных копий, сохраняемых в разных местах. Между копиями осуществляют ротацию. Например, в течение недели ежедневно копируют данные на носители резервного комплекта «А», а через неделю их заменяют комплектом «Б» и т. д.

Антивирусная защита программными средствами

Вспомогательными средствами защиты информации являются антивирусные программы и средства аппаратной защиты. Они предоставляют следующие возможности.

1. Создание «образа» жесткого диска на внешних носителях. Этот образ позволяет восстановить данные после повреждения системной области жесткого диска в результате вирусной атаки. Он позволяет также защитить данные от утраты при аппаратных сбоях или при неаккуратном форматировании жесткого диска.
2. Регулярное сканирование носителей данных в поисках компьютерных вирусов. Сканирование обычно выполняется автоматически при каждом включении компьютера или при подключении (размещении в читающем устройстве) нового носителя.

Следует иметь в виду, что антивирусная программа идентифицирует вирусы сравнением кода программ с кодами известных ей вирусов, хранящимися в базе данных. Если база устарела, а вирус является новым, сканер его не обнаружит. Для надежной работы следует регулярно обновлять антивирусную программу. Желательная периодичность — ежедневная. В неотвественных компьютерных системах допустимо еженедельное обновление.

1. Контроль за изменением размеров и других атрибутов файлов. Поскольку некоторые компьютерные вирусы на этапе размножения изменяют параметры зараженных файлов, контролирующая программа может обнаружить их деятельность и предупредить пользователя.
2. Контроль за обращениями к жесткому диску. В связи с тем, что наиболее опасные действия компьютерных вирусов обращены на модификацию данных, антивирусные программы способны идентифицировать подозрительную активность и предупредить пользователя о наличии угрозы.

Информационная безопасность в Интернете

В наиболее общей форме под информационной безопасностью понимается такой режим информационного обмена, при котором обеспечиваются конфиденциальность, целостность и доступность информации.

Конфиденциальность информации: условие информационного обмена, согласно которому доступ к информации обеспечивается только для авторизованных пользователей.

Доступность информации: требование к информационному обмену, согласно которому доступ авторизованных пользователей к информации и методам ее адекватного воспроизведения должен быть обеспечен по мере необходимости.

Целостность информации: требование достоверности и полноты информации, а также методов ее обработки в процессе информационного обмена.

Поскольку в Интернете передача данных осуществляется через открытые системы, исключить доступ к ним посторонних лиц невозможно даже теоретически. Соответственно, обеспечение режима информационной безопасности основывается не на ограничении доступа к данным, а на ограничении доступа к методам воспроизведения содержащейся в них информации. Одним из приемов такого ограничения является шифрование данных. Мы рассмотрим два вида шифрования данных: симметричное и несимметричное.

Симметричное шифрование информации

Системам симметричного шифрования столько же лет, сколько и письменному обмену информацией. Традиционный механизм шифрования заключается в том, что к документу применяется некий метод шифрования, основанный на использовании ключа, после чего документ становится недоступен для чтения обычными средствами. Его может прочитать только тот, кто знает ключ, то есть обладает адекватным методом воспроизведения информации.

Аналогичным образом происходит шифрование ответного сообщения. Если в процессе обмена информацией для шифрования и чтения используется один и тот же ключ, то такой криптографический процесс является симметричным.

Основное достоинство симметричного шифрования — высокая скорость работы алгоритмов. Это позволяет использовать симметричную криптографию для шифрования не только для коротких сообщений, но и объемных документов.

Дополнительные достоинства симметричного шифрования: высокая криптостойкость алгоритмов и их хорошая изученность. Первый фактор позволяет гарантировать, что ключ определенного размера не будет реконструирован в заданные сроки, а второй — то, что в эти сроки не будут созданы новые методы криптоанализа, радикально влияющие на стойкость алгоритма. Для примера в табл. 8.1. представлено количество операций, необходимых для реконструкции ключа.

Таблица 8.1. Количество операций, необходимых для реконструкции ключа

Длина ключа, бит	Количество операций
64	0,2*1020
128	0,3*1039
256	0,1*1078

При современном уровне развития вычислительной техники ключи длиной 56 бит реконструируются на специализированных рабочих станциях в течение нескольких недель. На реконструкцию ключей длиной 60 бит необходимы годы. В свою очередь, продолжительность реконструкции ключей размером 128 бит в миллиарды раз превышает возраст Вселенной.

Однако у симметричного шифрования есть не только достоинства, но и недостатки. Во-первых, чтобы два партнера могли начать защищенный информационный обмен, им необходимо передать ключ, а для этого опять-таки нужен защищенный канал связи. Таким образом, проблема повторяется, хотя и на другом уровне. Во-вторых, симметричное шифрование не позволяет однозначно аутентифицировать автора содержания, потому что ключом шифрования обладают несколько сторон.

Несимметричное шифрование

Недостатки симметричного шифрования успешно парируются несимметричными криптографическими системами, основанными на использовании не одного, а двух взаимосвязанных ключей. Такие системы нашли широкое применение в Интернете.

Ключи несимметричного шифрования образуют неразрывную пару. То, что зашифровано одним ключом, можно расшифровать и прочитать только с помощью второго ключа пары. Создатель ключей оставляет один ключ себе: этот ключ называется *закрытым*, или *личным*. Второй ключ широко распространяется среди партнеров, то есть публикуется. Его называют *открытым*, или *публичным ключом*. Например, публичный ключ компании может быть размещен на ее сайте. Формируя заказ, клиент зашифрует этим ключом данные о своей платежной карте. Прочитать их не сможет никто, кроме владельца закрытого ключа.

Напротив, когда компании потребуется связь с клиентом, она зашифрует свое сообщение закрытым ключом. Клиент прочитает послание, воспользовавшись имеющимся у него открытым ключом данной компании. При этом он будет совершенно уверен, от кого поступило сообщение, поскольку знает, кому принадлежит закрытый ключ.

Основные достоинства несимметричного шифрования соответствуют основным недостаткам шифрования симметричного. Во-первых, отпадает потребность в закрытом канале связи для передачи ключа, поскольку открытый ключ не представляет секрета. Во-вторых, открывается возможность для однозначной идентификации автора документа, потому что никто, кроме владельца закрытого ключа, не может подготовить документ, способный читаться открытым ключом.

Основной недостаток несимметричного шифрования — неудовлетворительная производительность программных средств, реализующих данную технологию. Производительности средств симметричного шифрования в сотни раз выше. Это связано с большим размером несимметричных ключей (он может достигать тысяч бит), а также с высокой вычислительной сложностью алгоритма. Поэтому технологии несимметричного шифрования принято использовать только для обмена короткими сообщениями.

Принцип достаточности защиты информации

Несимметричное шифрование существенно уступает симметричному по криптостойкости, если сравнивать их при одинаковом размере ключей. Поскольку один из ключей пары широко доступен, а алгоритм шифрования не является секретом, налицо теоретическая возможность для реконструкции закрытого ключа.

Тонкость заключается в том, что знание алгоритма и половинки ключа отнюдь не означает, что реконструкция закрытого ключа возможна в разумно приемлемые сроки с оправданными затратами средств. Так, например, правила игры в шахматы хорошо известны, и нетрудно создать алгоритм для перебора всех возможных шахматных партий, но он никому не нужен, поскольку даже самый быстрый современный суперкомпьютер будет работать над этой задачей дольше, чем существует жизнь на нашей планете.

Защита информации методами несимметричного шифрования не является абсолютной и не считается таковой. Но о ней можно говорить как о достаточной, если затраты на преодоление защиты существенно превышают ее ожидаемую ценность. В этом состоит *принцип достаточности защиты*, которым руководствуются при использовании несимметричных средств шифрования данных. То есть, выбирая уровень защиты информации, следует соотносить его с ценностью защищаемой информации.

Разумеется, не всегда реконструкцию закрытого ключа производят методами простого перебора комбинаций. Для этого существуют специальные методы, основанные на исследовании особенностей взаимодействия открытого ключа с определенными структурами данных. Область науки, посвященная этим исследованиям, называется *криптоанализом*, а средняя продолжительность времени, необходимого для реконструкции закрытого ключа по его открытому ключу, определяет *криптостойкость* алгоритма шифрования.

Для многих методов несимметричного шифрования криптостойкость, полученная в результате криптоанализа, существенно отличается от величин, заявляемых разработчиками алгоритмов на основании теоретических оценок. Поэтому во многих странах вопрос применения алгоритмов шифрования данных находится в поле законодательного регулирования. В частности, в России к использованию

в государственных органах разрешены только программные средства шифрования данных, прошедшие государственную сертификацию. Более того, деятельность по разработке и внедрению средств шифрования в Российской Федерации является лицензируемой и должна удовлетворять весьма строгим требованиям.

Гибридные методы защиты

Гибридные методы защиты информации основаны на сочетании двух основных технологий защиты. Они позволяют сочетать достоинства методов несимметричного шифрования с высокой производительностью симметричных методов. Рассмотрим особенности гибридных методов на примере создания защищенного канала связи по технологии *SSL* (Secure Sockets Layer).

1. Защищенное SSL-соединение устанавливается в два этапа. На первом этапе стороны обмениваются открытыми ключами и устанавливают первоначальный диалог на основе несимметричного шифрования. При этом стороны получают уверенность в том, с кем имеют дело, и в том, что на линии связи не произошла подмена партнера.
2. В ходе первоначального диалога стороны обмениваются сообщениями, на основании которых они могут совместно выработать общий симметричный ключ. Важно заметить, что симметричный ключ никуда не передается, поскольку у каждого из партнеров он вырабатывается непосредственно «на месте».
3. Полученный симметричный ключ является сеансовым, то есть он действителен только в одном сеансе связи. Пользуясь этим ключом, стороны могут обмениваться сообщениями, документами и любой иной информацией. Таким образом, между ними образуется защищенный канал связи. На рассмотренной ранее модели ISO/OSI этот канал связи относится к третьему уровню системы связи, к так называемому сеансовому уровню.

Подписание документа электронной цифровой подписью

Если сообщение (или его часть) зашифровано закрытым ключом, то получатель сообщения имеет возможность однозначно идентифицировать его создателя. Этот факт используется для создания такого атрибута электронных документов, как электронная цифровая подпись (ЭЦП). При этом дополнительно возникает возможность убедиться в целостности (полноте и неизменности) содержания документа. Весь процесс подписания документа с помощью ЭЦП происходит в несколько этапов.

1. На первом этапе документ обрабатывается математически с помощью некоторой стандартной функции. Ее называют хэш-функцией, а процедуру обработки документа называют хэшированием. В результате хэширования образуется так называемый слепок документа. Это последовательность данных сравнительно небольшого размера (несколько десятков байтов). Иногда ее также называют оттиском документа. Фактически эта последовательность выполняет функции электронной печати. Она заверяет целостность документа. Если в ходе транспортировки в документ будут внесены какие-либо изменения, то оттиск пере-

станет соответствовать содержанию документа, и адресат легко установит факт подлога.

2. На втором этапе полученный оттиск шифруется закрытым ключом отправителя. Поскольку несимметричное шифрование — процесс трудоемкий, в гражданском документообороте нет смысла шифровать все содержание документа, вполне можно ограничиться шифрованием одного только оттиска.
3. Далее к документу прилагается открытый ключ, позволяющий прочесть сообщение и удостовериться в идентичности партнера и аутентичности содержания.
4. Чтобы избежать подмены открытого ключа на маршруте движения документа, этот ключ тоже шифруется, но теперь уже закрытым ключом специальной уполномоченной организации, которая называется удостоверяющим центром. В обязанности удостоверяющих центров входит подтверждение открытых ключей.
5. В заключение, к документу прилагается открытый ключ удостоверяющего центра. При этом образуется связка ключей. Эту связку можно рассматривать как сертификат ключа подписи (СКП), выданный удостоверяющим центром.

(СКП) сертификат ключа подписи — электронный документ с электронной цифровой подписью уполномоченного лица удостоверяющего центра, который включает в себя открытый ключ электронной цифровой подписи участника информационной системы и выдается ему удостоверяющим центром для подтверждения подлинности электронной цифровой подписи, идентификации владельца сертификата ключа подписи и определения его полномочий.

Чтение документа, подписанного электронной цифровой подписью

Чтение документов, подписанных ЭЦП, происходит в порядке, обратном их подписанию. При этом автоматически выполняется проверка всех элементов ЭЦП. Происходит это прозрачно для пользователя. Он получает предупреждение только при обнаружении каких-либо несоответствий.

1. Сначала получатель сообщения проверяет сертификат ключа подписи ЭЦП и устанавливает, кем он выдан, когда и на какой срок.
2. Если в сертификате указаны адекватные данные, происходит чтение открытого ключа отправителя. Напомним, что он зашифрован закрытым ключом удостоверяющего центра, но наличие в сертификате соответствующего открытого ключа позволяет выполнить чтение.
3. С помощью открытого ключа отправителя выполняется чтение оттиска документа. Если оттиск читается нормально, значит, открытый ключ отправителя соответствует его закрытому ключу, и личность отправителя можно считать установленной.
4. На последнем этапе проверяется целостность документа. С помощью хэш-функции вычисляется оттиск документа и сравнивается с оттиском, полученным при чтении ЭЦП. Если оба оттиска совпадают, значит, целостность документа не нарушена.

Практическое занятие

Упражнение 8.1. Создание соединения удаленного доступа

1. Откройте окно Установка подключения или сети: Пуск ► Панель управления ► Сеть и интернет ► Центр управления сетями и общим доступом ► Настройка нового подключения или сети.
2. Выберите тип подключения – Коммутируемое. Щелкните на кнопке Далее.
3. На следующем этапе работы введите в появившемся окне (рис. 8.3) данные, необходимые для создания подключения. Эти данные обычно предоставляются поставщиком услуг коммутируемого доступа при заключении договора. Если подключение осуществляется по предоплаченной интернет-карте, данные о номере телефона провайдера, а также об имени и пароле можно найти на карте и в прилагаемой к ней инструкции.

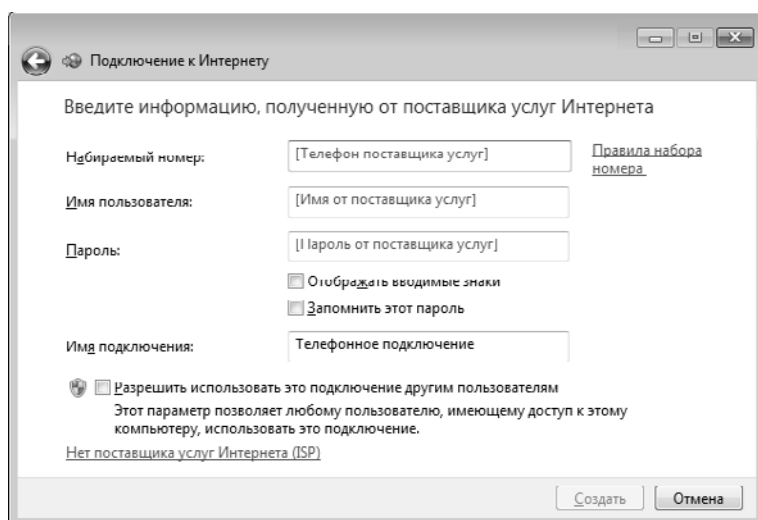


Рис. 8.3. Настройка свойств коммутируемого подключения к Интернету

Сохранять информацию о пароле (флажок Запомнить этот пароль) можно только на компьютерах, находящихся в личном пользовании. На компьютерах, предназначенных для коллективного использования, эту информацию не сохраняют. В операционных системах семейства Windows защита конфиденциальных данных организована не идеально. Подготовленному пользователю доступны косвенные данные, дающие возможности извлечь зашифрованные сведения обходными приемами.

4. Заполните поле Набираемый номер. Щелкните по ссылке Правила набора номера. По умолчанию параметр Тип набора номера установлен в Тоновый. Такой способ набора номера используется на большинстве современных АТС. Если тоновый набор на вашей АТС не работает, установите переключатель Тип набора номера в состояние Импульсный.
5. Укажите имя пользователя и (дважды) пароль доступа.

6. Введите произвольное название нового соединения в поле **Имя подключения**. Щелкните на кнопке **Создать**.

В некоторых случаях имя пользователя и пароль предоставляются на веб-сайте поставщика услуг в ходе интерактивной регистрации. В этом случае для первичного подключения к Интернету и выполнения регистрации поставщик предоставляет имя пользователя и пароль для бесплатного гостевого доступа. В этом упражнении допустимо создать такое соединение для гостевого доступа к сайту поставщика услуг.

Упражнение 8.2. Настройка подключения для удаленного доступа и установление соединения

1. Щелкните мышью значок сети на Панели задач, в появившемся окне выполните команду **Свойства** контекстного меню для нужного подключения (в нашем случае это – подключение **Телефонное подключение**, созданное в предыдущем упражнении). Откроется окно.
2. На вкладке **Общие** проверьте правильность ввода телефонного номера поставщика услуг Интернета и правильность выбора и настройки модема. В случае необходимости внесите необходимые изменения.

Если поставщик услуг Интернета предоставил несколько телефонных номеров для подключения к его серверу, щелкните на кнопке **Другие** и введите дополнительные номера телефонов, используя кнопку **Добавить**.

3. На вкладке **Сеть** в списке **Компоненты**, используемые этим подключением флажки **Служба доступа к файлам и принтерам сетей Microsoft** и **Клиент для сетей Microsoft** должны быть сброшены из соображений безопасности. Выберите в этом списке пункт **Протокол Интернета версии 4 (TCP/IPv4)** и щелкните на кнопке **Свойства**.
4. Включите переключатель ввода IP-адреса в соответствии с указаниями поставщика услуг (для коммутируемого соединения обычно включают переключатель **Получить IP-адрес автоматически**).
5. Введите адреса серверов DNS. Если эти адреса получены от поставщика услуг, включите переключатель **Использовать следующие адреса DNS-серверов** и введите по четыре числа для первичного и вторичного серверов DNS. Если адреса не получены, возможно, что они вводятся автоматически. В этом случае включите переключатель **Получить адрес DNS-сервера автоматически**.
6. Щелчком на кнопке **ОК** закройте диалоговое окно настройки свойств протокола TCP/IP.
7. Щелчком на кнопке **ОК** закройте диалоговое окно настройки свойств подключения.
8. Для установления соединения с сервером поставщика услуг щелкните мышью по значку сети на Панели задач и щелкните по названию подключения. При необходимости, если этого не было сделано ранее, введите в поля появившегося окна имя пользователя и пароль. Запустите процесс установления соединения нажатием на соответствующую кнопку.

Глава девятая

Получение информации из Интернета

9.1. Основные понятия World Wide Web

Сегодня Интернет используется как источник разносторонней информации по различным областям знаний. Большинство документов, доступных на серверах Интернета, имеют *гипертекстовый формат*. Службу Интернета, управляющую передачей таких документов, называют *World Wide Web* (Web, WWW). Этим же термином, или *средой WWW*, называют обширную совокупность веб-документов, между которыми существуют гипертекстовые связи.

Среда WWW не имеет централизованной структуры. Она пополняется теми, кто желает разместить в Интернете свои материалы, и может рассматриваться как *информационное пространство*. Как правило, документы WWW хранятся на постоянно подключенных к Интернету компьютерах — *веб-серверах*. Обычно на веб-сервере размещают не отдельный документ, а группу взаимосвязанных документов. Такая группа представляет собой *веб-узел* (альтернативный термин — *веб-сайт*). Размещение подготовленных материалов на веб-узле называется *веб-изданием* или *веб-публикацией*.

Веб-страница

Отдельный документ World Wide Web называют *веб-страницей*. Обычно это комбинированный документ, который может содержать текст, графические иллюстрации, мультимедийные и другие вставные объекты. Для создания веб-страниц используется язык *HTML* (HyperText Markup Language — язык разметки гипертекста), который при помощи вставленных в документ *тегов* описывает логическую структуру документа, управляет форматированием текста и размещением вставных объектов. *Интерактивные веб-узлы* получают информацию от пользователя через *формы* и генерируют запрошенную веб-страницу с помощью специальных программ (сценариев CGI, PHP-скриптов), динамического HTML и других средств.

Гиперссылки

Отличительной особенностью среды World Wide Web является наличие средств перехода от одного документа к другому, тематически с ним связанному, без явного указания адреса. Связь между документами осуществляется при помощи *гипертекстовых ссылок* (или просто гиперссылок). Гиперссылка — это выделенный фрагмент документа (текст или иллюстрация), с которым ассоциирован адрес другого веб-документа. При использовании гиперссылки (обычно для этого требуется навести на нее указатель мыши и один раз щелкнуть) происходит переход по гиперссылке — открытие веб-страницы, на которую указывает ссылка. Механизм гиперссылок позволяет организовать тематическое путешествие по World Wide Web без использования (и даже без знания) адресов конкретных страниц. Для

записи адресов документов Интернета (веб-страниц) используется форма, называемая *адресом URL*, рассмотренная в предыдущей главе.

Средства просмотра Web

Документы Интернета предназначены для отображения в *электронной форме*, причем автор документа не знает, каковы возможности компьютера, на котором документ будет отображаться. Поэтому язык HTML обеспечивает не столько форматирование документа, сколько описание его логической структуры. Форматирование и отображение документа на конкретном компьютере производится специальной программой — *браузером* (от англ. *browser*).

Основные функции браузеров следующие:

- ❑ установление связи с веб-сервером, на котором хранится документ, и загрузка всех компонентов комбинированного документа;
- ❑ интерпретация тегов языка HTML, форматирование и отображение веб-страницы в соответствии с возможностями компьютера, на котором браузер работает;
- ❑ предоставление средств для отображения мультимедийных и других объектов, входящих в состав веб-страниц, а также механизма расширения, позволяющего настраивать программу на работу с новыми типами объектов;
- ❑ обеспечение автоматизации поиска веб-страниц и упрощение доступа к веб-страницам, посещенным ранее;
- ❑ предоставление доступа к встроенным или автономным средствам для работы с другими службами Интернета.

9.2. Работа с программой Internet Explorer 8

Со стороны Интернета работу службы World Wide Web обеспечивают серверные программные средства — веб-серверы. Со стороны пользователя работа обеспечивается клиентскими программами — веб-браузерами. Существует множество разных браузеров, выпускаемых разными компаниями.

В принципе, все браузеры выполняют одни и те же функции, и выбор конкретного средства просмотра — дело вкуса и привычки пользователя. Однако у браузера Microsoft Internet Explorer есть преимущество перед остальными, заключающееся в том, что, начиная с операционной системы Windows 98, он поставляется вместе с системой и интегрирован в нее так, что является ее неотъемлемым компонентом.

С последней версией операционной системы Windows 7 поставляется версия браузера Internet Explorer 8.0 (IE 8). Основная роль IE 8 заключается в обеспечении доступа к ресурсам корпоративных intranet-сетей и к работе с WWW. Кроме того, браузер имеет средства для работы с FTP-ресурсами и некоторые другие функции.

Для запуска браузера Internet Explorer можно использовать значок Internet Explorer на Рабочем столе или на Панели быстрого запуска, а также Главное меню

(Пуск ► Все программы ► Internet Explorer). Кроме того, программа запускается автоматически при попытке открыть документ Интернета или локальный документ в формате HTML. Windows можно настроить на использование другого браузера по умолчанию. В таком случае для тех же самых функций будет использоваться именно выбранный браузер.

Если соединение с Интернетом отсутствует, то после запуска программы на экране появится диалоговое окно для управления установкой соединения. При невозможности установить соединение сохраняется возможность просмотра в автономном режиме ранее загруженных веб-документов. При наличии соединения после запуска программы на экране появится так называемая «домашняя», или основная, страница, выбранная при настройке программы.

Открытие и просмотр веб-страниц

Просматриваемая веб-страница отображается в рабочей области окна. По умолчанию воспроизводится все ее содержимое, включая графические иллюстрации и встроенные мультимедийные объекты. Управление просмотром осуществляется при помощи строки меню, панелей инструментов, а также активных элементов, имеющих в открытом документе, например гиперссылки.

Если URL-адрес веб-страницы известен, его можно ввести в поле панели Адрес и щелкнуть на кнопке Переход. Страница с указанным адресом открывается вместо текущей. Наличие средства автозаполнения адресной строки упрощает повторный ввод адресов. Вводимый адрес автоматически сравнивается с адресами ранее просматривавшихся веб-страниц. Все подходящие адреса отображаются в раскрывающемся списке панели Адрес. Если нужный адрес есть в списке, его можно выбрать клавишами ВВЕРХ и ВНИЗ, после чего щелкнуть на кнопке Переход. При отсутствии нужного адреса ввод продолжают как обычно.

Работа с гиперссылками

Навигация по Интернету чаще выполняется не путем ввода адреса URL, а посредством использования гиперссылок. При отображении веб-страницы на экране гиперссылки выделяются цветом (обычно синим) и подчеркиванием. Обычно подчеркивание применяют *только* для выделения гиперссылок. Более надежным признаком является форма указателя мыши. При наведении на гиперссылку он принимает форму кисти руки с вытянутым указательным пальцем, а сама гиперссылка при соответствующей настройке браузера изменяет цвет. Адрес URL, на который указывает ссылка, отображается в строке состояния. При щелчке на гиперссылке соответствующая веб-страница загружается вместо текущей. Если гиперссылка указывает на произвольный файл, происходит его загрузка.

На веб-страницах могут также встречаться графические ссылки (то есть гиперссылки, представленные рисунком) и изображения-карты, объединяющие несколько ссылок в рамках одного изображения. Для просмотра ссылок на открытой веб-странице удобно использовать клавишу TAB. При нажатии этой клавиши фокус ввода (пунктирная рамка) перемещается к следующей ссылке или к следующему элементу управления, отображаемому на странице. Это может быть кнопка, поле

ввода, флажок. Перейти по ссылке (или активировать элемент управления) можно, нажав клавишу ENTER. При таком подходе последовательно перебираются текстовые и графические ссылки, а также отдельные области изображений-карт.

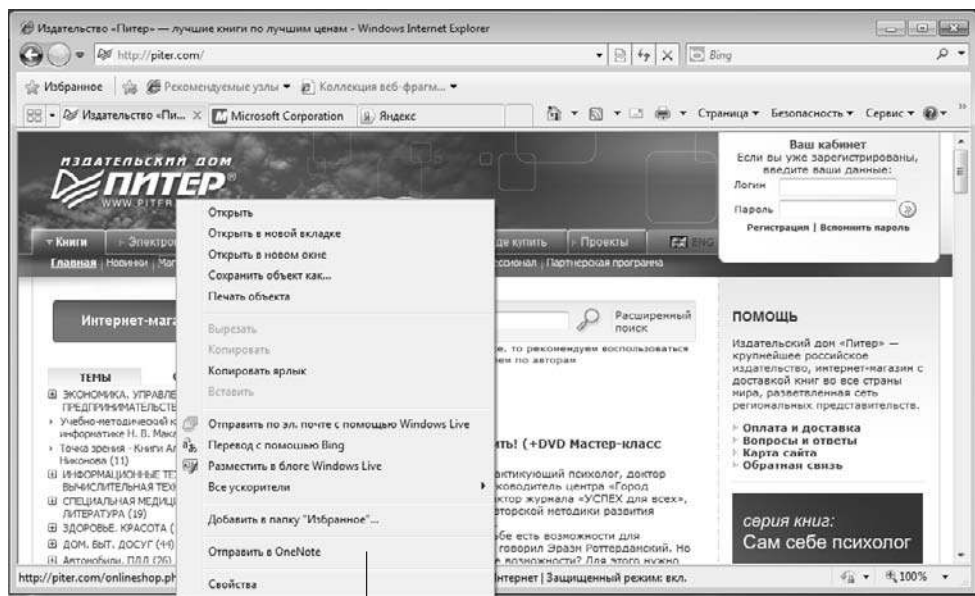
Дополнительные возможности использования гиперссылок предоставляет их контекстное меню. Чтобы открыть новую страницу, не закрывая текущей, применяют команду Открыть в новой вкладке. В одном окне программы может быть открыто множество вкладок. Вкладки значительно упрощают работу со множеством открытых окон.

В том случае, если нужно открыть новую страницу в новом окне браузера, можно воспользоваться командой Открыть в новом окне. В результате открывается новое окно браузера.

Адрес URL, заданный ссылкой, можно поместить в буфер обмена при помощи команды Копировать ярлык. Его можно вставить в поле панели Адрес или в любой другой документ для последующего использования.

Другие операции, относящиеся к текущей странице и ее элементам, также удобно осуществлять через контекстное меню. Так, например, рисунок, имеющийся на странице, можно:

- ☐ сохранить как файл (Сохранить изображение как, Сохранить объект как);
- ☐ использовать как фоновый рисунок (Сделать фоновым рисунком).



Контекстное меню объекта

Рис. 9.1. Веб-страница в ходе просмотра

Если рисунок выполняет функции графической ссылки, к нему можно применять как команды, относящиеся к изображению, так и команды, относящиеся к ссылке.

Приемы управления браузером

Необходимость определенных действий в ходе просмотра документов World Wide Web часто диктуется самим ходом работы. В таких случаях удобно использовать кнопки, расположенные около адресной строки. Для того чтобы вернуться к странице, которая просматривалась некоторое время назад, используют кнопку **Назад**. Чтобы возвратиться на несколько страниц назад, можно использовать присоединенную к ней кнопку раскрывающегося списка. Отменить действия, выполненные при помощи кнопки **Назад**, позволяет кнопка **Вперед**.

Если процесс загрузки страницы затянулся или надобность в ней отпала, используют кнопку адресной строки **Остановить**. Заново загрузить веб-страницу, если ее загрузка была прервана или содержание документа изменилось, позволяет кнопка **Обновить**. Чтобы немедленно загрузить «домашнюю» (основную) страницу, с которой браузер обычно начинает работу, пользуются кнопкой **Домой**, которая расположена в **Командной строке**.

Создать новое окно, сохранить открытый документ на своем компьютере, распечатать его, включить или выключить режим автономной работы позволяют команды меню **Страница** и **Сервис Командной строки**.

Как и большинство программ Windows 7, Internet Explorer 8 имеет классическую строку меню, скрытую по умолчанию. Для того чтобы строка меню появилась на экране, нужно нажать клавишу ALT или F10 на клавиатуре. Подавляющее большинство действий в IE 8 можно произвести без использования строки меню.

Копирование фрагментов документа в буфер обмена, поиск текста на веб-странице осуществляются при помощи команд классического меню **Правка** или, что удобнее, с использованием контекстного меню.

Включение и выключение отображения служебных элементов окна (панелей инструментов, дополнительных панелей, строки состояния), выбора шрифта и кодировки символов осуществляются через меню **Сервис Командной строки**.

Ведение списка регулярно посещаемых страниц и быстрый доступ к ним осуществляются через инструменты **Панели избранного**.

Переход к использованию программ для работы с другими службами Интернета, а также настройка браузера осуществляются через меню **Командной строки Сервис**.

Работая в IE 8, можно пользоваться так называемыми ускорителями. Для того чтобы воспользоваться одним из ускорителей, достаточно выделить объект веб-страницы. Например, это может быть несколько слов из текста, фрагмент страницы, изображение. После того как объект будет выделен, около него появится значок, щелкнув на котором левой кнопкой мыши можно увидеть меню ускорителей (рис. 9.2). Команды этого меню дублируются в обычном контекстном меню, вызванном для фрагмента, однако работать с выделенным меню ускорителей удобнее.

Среди стандартных ускорителей можно отметить, например, быструю отправку выделенного фрагмента текста в поисковую систему Bing для последующего поиска по этому фрагменту текста.

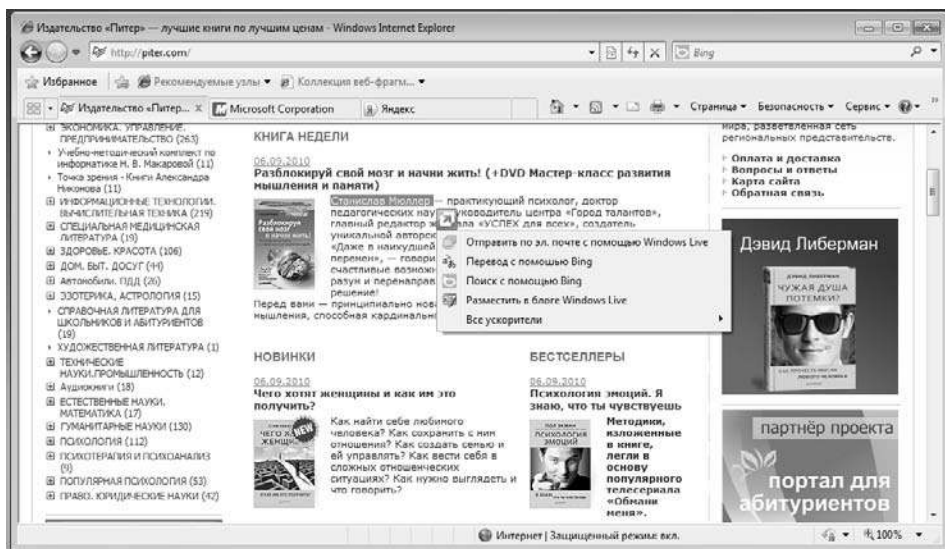


Рис. 9.2. Использование ускорителей

Настройка свойств браузера

Для эффективной и комфортной работы в Интернете необходима настройка браузера. Параметры оптимальной настройки зависят от многих факторов:

- ☐ свойств видеосистемы компьютера;
- ☐ производительности действующего соединения с Интернетом;
- ☐ содержания текущего веб-документа;
- ☐ личных предпочтений индивидуального пользователя.

Начать настройку программы Internet Explorer можно как из самой этой программы (Сервис ► Свойства обозревателя), так и через общесистемное средство Windows — Панель управления ► Сеть и Интернет ► Свойства обозревателя.

Общие параметры работы браузера задают на вкладке **Общие** (рис. 9.3). Здесь можно указать, какую страницу следует использовать в качестве основной, задать объем дискового пространства для хранения временных файлов Интернета и удалить такие временные файлы, а также страницы, подготовленные для чтения в автономном режиме. Правила хранения временных файлов задаются с помощью кнопки **Параметры**. Чем реже программа проверяет соответствие версий давно загруженной страницы и реального документа, тем больше экономится времени на загрузке страниц, но увеличивается риск их устаревания. Кнопка **Обновить** (справа от строки браузера **Адрес**) позволит получить самую последнюю версию документа независимо от настроек.

Управление оформлением отображаемых веб-страниц также осуществляется элементами управления вкладки **Общие**. Используемые цвета настраиваются при

помощи кнопки Цвета, а шрифты — при помощи кнопки Шрифты. Эти настройки подчинены тому, что задано в самом веб-документе.

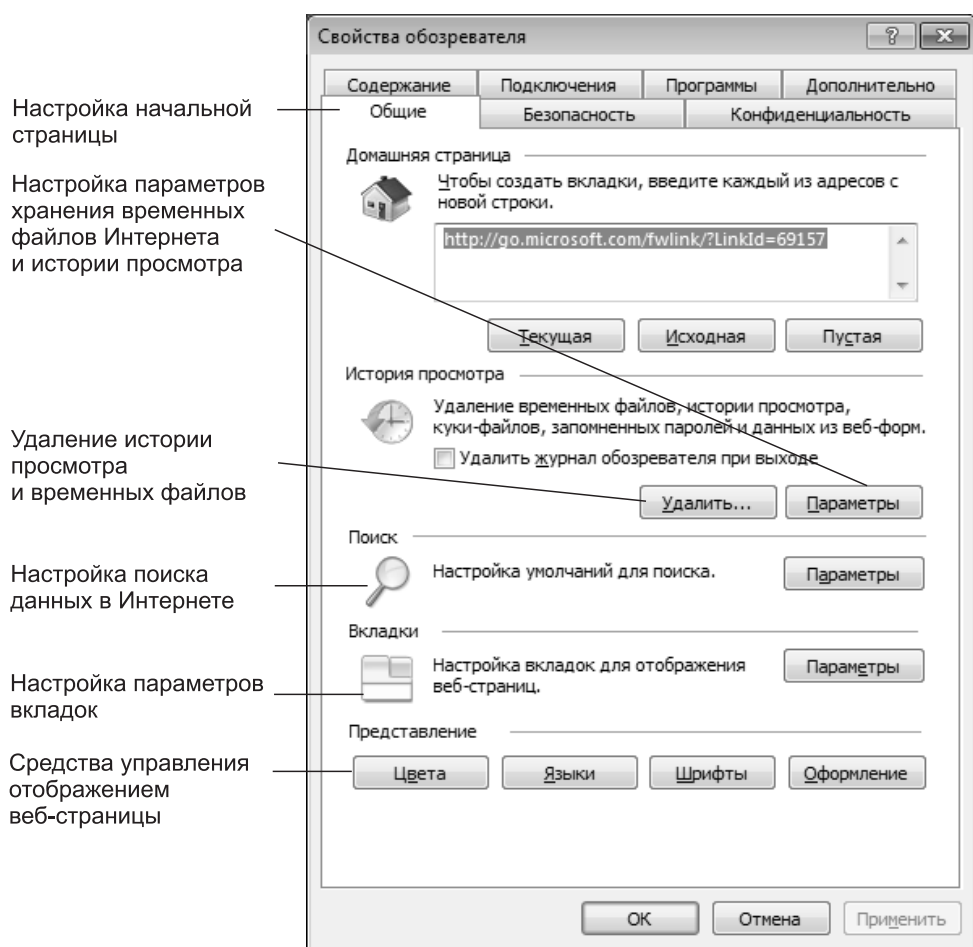


Рис. 9.3. Управление основными параметрами отображения веб-страниц

Если по какой-либо причине необходим полный контроль над оформлением отображаемых документов, используют кнопку Оформление. С ее помощью можно задать принудительное использование параметров форматирования, заданных в свойствах браузера. Это может относиться к используемым цветам (флажок Не учитывать цвета, указанные на веб-страницах), начертаниям шрифтов (Не учитывать шрифты, указанные на веб-страницах) и размерам шрифтов (Не учитывать размеры шрифтов, указанные на веб-страницах).

Настройка свойств соединения с Интернетом осуществляется при помощи вкладки Подключения. Здесь доступны те же операции, что и при непосредственном использовании папки Сетевые подключения. Кроме того, можно указать, какое именно соединение должно использоваться при работе браузера. С помощью

переключателей можно задать режим отказа от автоматического подключения, стандартный режим подключения при отсутствии соединения или режим использования только одного соединения.

Выбор программ, используемых для работы в Интернете, осуществляется с помощью вкладки Программы.

Средства защиты от потенциально опасного содержимого веб-документов предоставляет вкладка Безопасность. Она позволяет указать веб-узлы, взаимодействие с которыми следует считать опасным, и запретить прием с них информации, которая может оказаться разрушительной.

Вкладка Конфиденциальность позволяет ограничить доступ веб-узлов к личной информации, хранимой на компьютере. Веб-узлы имеют потенциальную возможность сохранять на компьютере пользователя небольшие информационные файлы («маркеры cookie») и читать их при последующих обращениях к тому же узлу. На вкладке Конфиденциальность можно задать ограничения на прием маркеров cookie.

Для ограничения доступа к узлам с неприемлемым содержанием, а также для управления использованием электронных сертификатов служат элементы управления вкладки Содержание.

Прочие настройки сосредоточены на вкладке Дополнительно. Они позволяют:

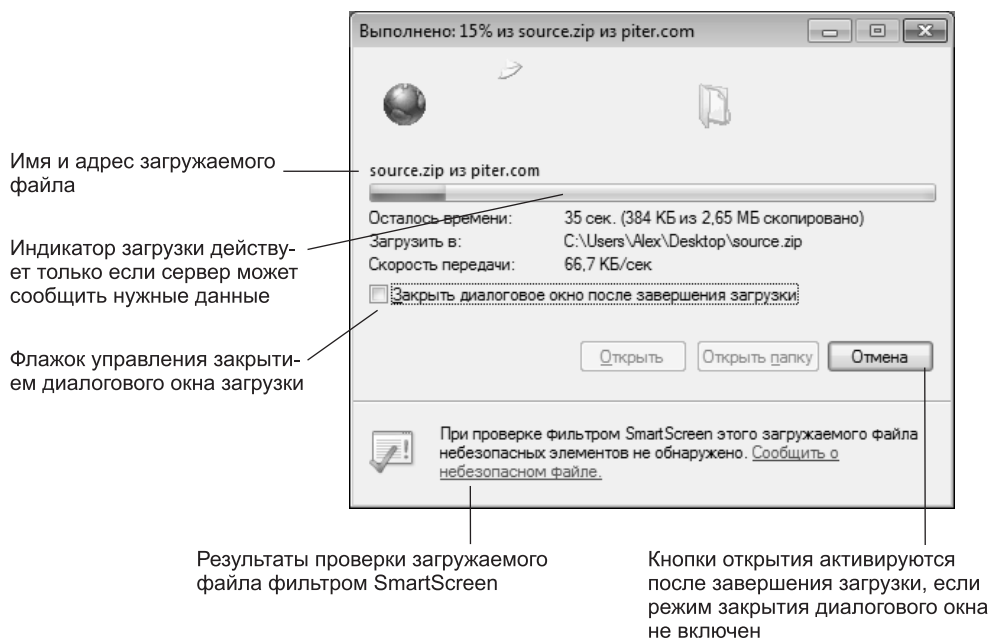
- ☐ соблюдать конфиденциальность работы с помощью средств шифрования, использования электронных сертификатов и своевременного удаления временных файлов;
- ☐ контролировать использование средств языка Java;
- ☐ управлять отображением мультимедийных объектов;
- ☐ использовать дополнительные настройки оформления;
- ☐ управлять режимом поиска веб-страниц, содержащих нужную информацию.

Прием файлов из Интернета

Гиперссылки, имеющиеся на веб-страницах, могут указывать на документы разных типов. Если браузер не способен отображать файлы определенного типа (например, исполняемые файлы с расширением .EXE, архивы .ZIP и пр.), инициируется процесс загрузки данного файла на компьютер.

Программа Internet Explorer 8.0 запускает мастер загрузки файла, на первом этапе работы которого требуется указать, следует ли открыть файл или сохранить его на диске. «Открытие» файла подразумевает загрузку его в каталог временных файлов и немедленный запуск (если это исполняемый файл) или открытие с помощью программы, которая предназначена для работы с файлами этого типа. Такой подход открывает путь на компьютер для небезопасной информации. Надежнее выбрать сохранение файла на диске. В этом случае требуется выбрать папку, в которой следует сохранить файл, и задать имя файла.

Ход загрузки файла отображается в специальном окне (рис. 9.4). Шкала хода работы появляется только в том случае, когда мастер управления загрузкой может получить информацию о полной длине файла, а это возможно только когда файл загружается непосредственно с веб-узла.

**Рис. 9.4.** Загрузка файла

При загрузке файла с узла FTP такие данные предоставляются не всегда. За ходом загрузки можно также следить по строке заголовка окна или, если окно свернуто или скрыто другими окнами, по надписи на кнопке Панели задач. Процесс загрузки файла не препятствует параллельному просмотру веб-страниц или другим операциям в Интернете.

После окончания загрузки окно загрузки закрывается автоматически, если установлен флажок **Закрывать диалоговое окно после завершения загрузки**. В противном случае после окончания загрузки активируются кнопки **Открыть** и **Открыть папку**, которые позволяют соответственно открыть только что загруженный файл или папку, которая его содержит.

Загрузку файла можно прервать в любой момент при помощи кнопки **Отмена**. После прерывания загрузки пользователем или вследствие разрыва соединения эту операцию необходимо начать заново. В операционной системе Windows 7 нет средств, способных возобновить загрузку файла, прерванную по какой-либо причине. Это возможно только при использовании специальных служебных программ (менеджеров загрузки).

Файлы, доступные для загрузки любым пользователям, чаще всего хранятся на FTP-узлах. Для доступа к FTP-узлу можно указать его адрес URL на панели **Адрес**. Браузер Internet Explorer 8.0 обеспечивает по умолчанию *анонимное подключение* к узлу FTP, при котором разрешены только просмотр каталогов и загрузка файлов. Если анонимный доступ не разрешен, на экране отображается диалоговое окно для ввода имени и пароля (разумеется, их следует знать).

Для загрузки файлов с FTP-узла можно воспользоваться теми же приемами, которые применяются при загрузке файлов с обычных сайтов. Так, при щелчке

по гиперссылке, которая представляет имя файла, будет выдано стандартное окно загрузки файлов.

Если вы предполагаете много работать с FTP-сервисами, лучше всего использовать специальный FTP-менеджер. Такие программы значительно облегчают взаимодействие с FTP-серверами.

9.3. Поиск информации в World Wide Web

Интернет имеет три функции: *коммуникационную, информационную и управленческую*. Разные службы могут обеспечивать разные функции. Хотя в рамках службы World Wide Web есть сервисы, исполняющие коммуникационные и управленческие функции, основное назначение этой службы — информационное. Когда нам нужно разыскать какие-то сведения, мы обращаемся за данными в первую очередь в информационное пространство Web.

Это пространство отличается гигантскими размерами и содержит миллиарды веб-документов. Найти среди них именно то, что нужно, — это особая, отнюдь не простая задача. Разумеется, можно пользоваться рекомендациями знакомых, коллег по работе, адресами URL, опубликованными в средствах массовой информации, но службе WWW совершенно необходимы свои поисковые сервисы, и они существуют.

Поисковая система представляет собой специализированный веб-узел. Пользователь сообщает поисковой системе данные о содержании искомой веб-страницы, а поисковая система выдает список гиперссылок на страницы, на которых упоминаются соответствующие сведения. Существует несколько моделей, на которых основана работа поисковых систем, но исторически две модели приобрели наибольшую популярность — это *поисковые каталоги* и *поисковые указатели*.

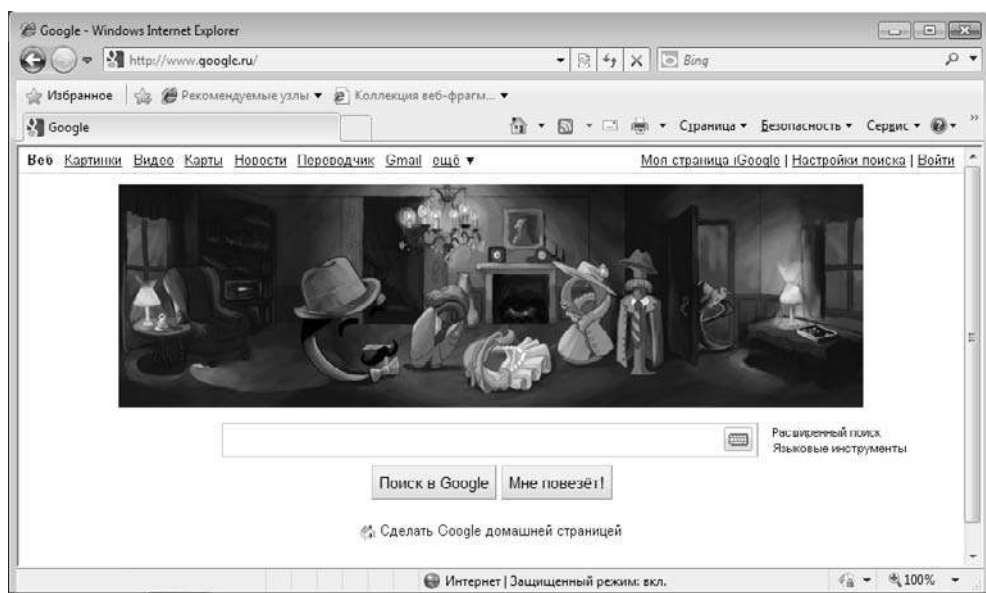


Рис. 9.5. Поисковая система Google

История поисковых систем началась с поисковых каталогов. В процессе роста Интернета, увеличения количества веб-сайтов и документов на первые позиции вышли поисковые указатели.

Современные поисковые системы (рис. 9.5) обычно построены по принципам поискового указателя. Нередко они включают в себя и поисковые каталоги, которым отводится второстепенная роль.

Поисковые каталоги

Поисковые каталоги устроены по тому же принципу, что и тематические каталоги крупных библиотек. Обратившись к поисковому каталогу, мы находим на его основной странице сокращенный список крупных тематических категорий, например таких, как Наука (Science), как показано на примере поискового каталога dmoz.org (рис. 9.6).

Каждая запись в списке категорий — это гиперссылка. Щелчок на ней открывает следующую страницу поискового каталога, на котором данная тема представлена подробнее, например по предметам: Астрономия, Биология, География, Математика, Физика и многие другие. Щелчок на названии темы (например, Физика) открывает страницу со списком разделов (Астрофизика, Атомная физика, Гидродинамика, Механика и т. д.). Продолжая погружение в тему, можно дойти до списка конкретных веб-страниц и выбрать себе тот ресурс, который лучше подходит для решения задачи.

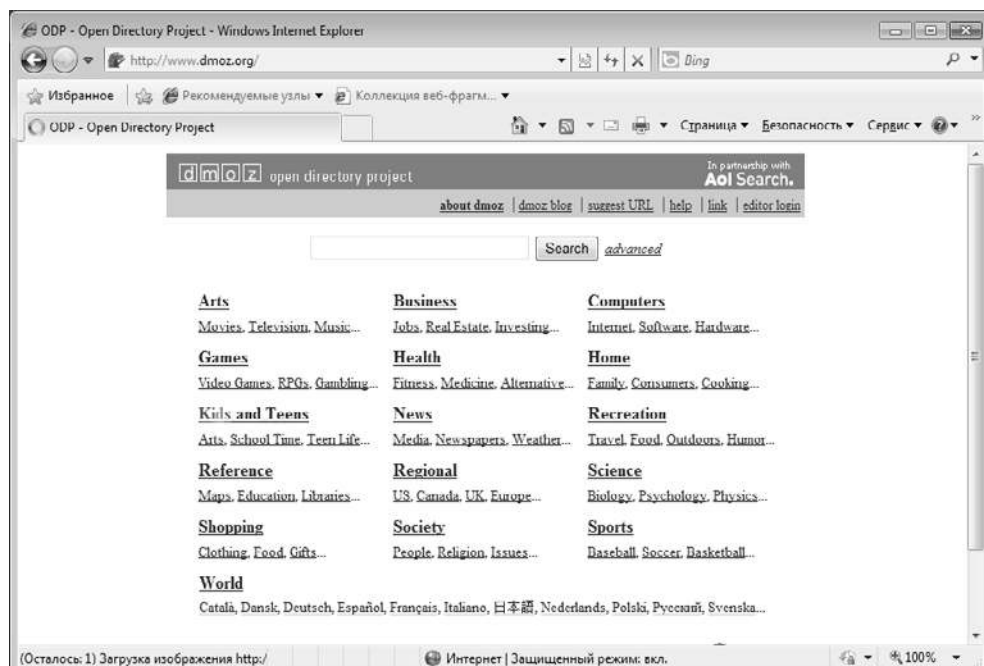


Рис. 9.6. Основная страница поискового каталога dmoz.org

Работа с поисковыми каталогами интуитивно проста. В них поиск информации практически всегда завершается более или менее плодотворно. Однако за этой простотой скрывается высочайшая сложность создания и ведения каталога. Поисковые каталоги создаются вручную, коллективом высококвалифицированных редакторов. При этом общий объем каталогизированных веб-ресурсов невелик, а степень охвата общего объема ресурсов WWW непрерывно уменьшается.

Несмотря на низкий коэффициент охвата, поисковые каталоги все еще пользуются определенной популярностью. Их принято использовать для первичного, реферативного поиска информации по заданной теме. Если для пользователя тема является совершенно новой и неисследованной, то ему, прежде всего, нужны указатели на классические, наиболее содержательные ресурсы, а именно это и обеспечивают поисковые каталоги. Человеческий фактор, связанный с тем, что над составлением каталога работают люди, а не программы, обеспечивает качественный отбор наиболее важных ресурсов по каждой теме.

Для поисковых каталогов вопрос несоответствия между размерами исследованного и неисследованного веб-пространства стоит особенно остро. Одним из направлений развития поисковых каталогов является внедрение SMART-технологий автоматической каталогизации.

Существует множество теоретических изысканий в области SMART-технологий, но наиболее перспективной является модель векторного информационного пространства. Представим себе эксперта в какой-то области, например в физике. Если ему поставить задачу, то, наверное, он сможет составить словари, характерные для таких областей, как Механика, Термодинамика, Оптика и т. п. Проанализировав множество документов, относящихся к этим научным областям, он сможет не только указать характерные термины и понятия, но и дать им весовые оценки. Так, например, достаточно очевидно, что слово «перемещение» имеет больший вес в механике, чем в термодинамике. Комбинируя термины и весовые коэффициенты, можно строить многомерные системы координат, в которых различные области знания описывались бы разными многомерными векторами.

Автоматически получив новую веб-страницу, поисковая система может построить для нее математический вектор, основанный на формальном анализе содержания. Сравнивая этот вектор с уже рассчитанными векторами для различных областей знания, система может без участия человека предположить, к какой категории, теме и разделу относится тот или иной документ.

При таком подходе не обязательно хранить копии всех известных веб-страниц, как не надо хранить и их поисковые указатели. Вполне достаточно для каждого веб-документа хранить лишь его URL-адрес и число, соответствующее вектору.

Поисковые указатели

Автоматическую каталогизацию веб-ресурсов и удовлетворение запросов клиентов выполняют так называемые *поисковые указатели*. Из процесса наполнения базы данных поисковой системы исключается человеческий фактор. При этом падает качество ссылок, предоставляемых системой по результатам поиска, но одновременно увеличивается их количество.

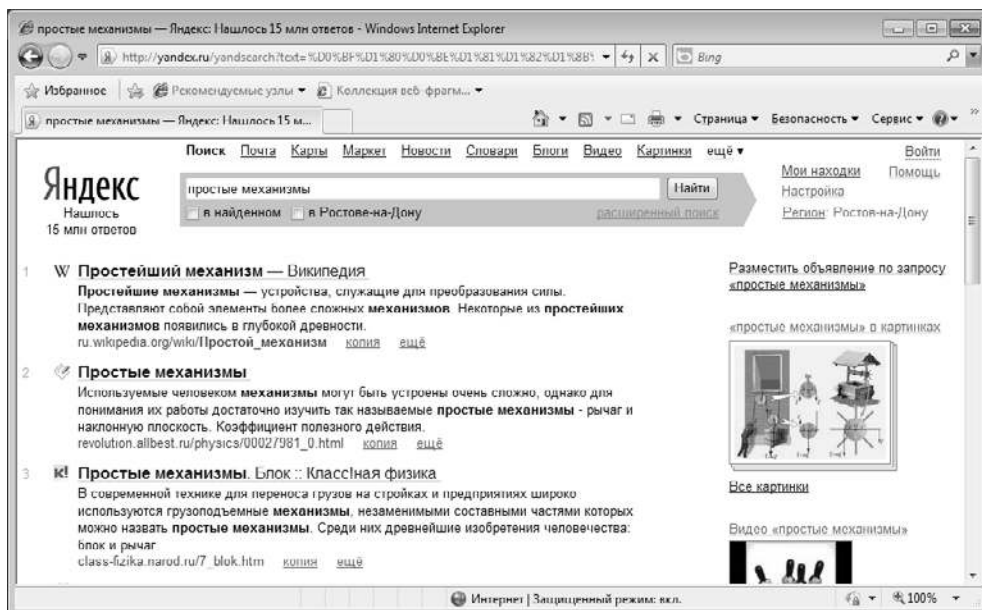


Рис. 9.7. Поиск информации по ключевым словам с помощью поисковой системы Яндекс

Основной принцип работы поискового указателя заключается в поиске веб-ресурсов по *ключевым словам*. Пользователь описывает искомый ресурс с помощью ключевых слов, после чего дает задание на поиск. Поисковая система анализирует данные, хранящиеся в своей базе, и выдает список веб-страниц, соответствующих запросу. Вместе с гиперссылками выдаются краткие сведения о найденных ресурсах, на основании которых пользователь может выбрать нужные ему ресурсы (рис. 9.7).

Разные поисковые указатели применяют разные информационные технологии для обработки запросов пользователей. Чтобы эффективно выполнять поиск информации в WWW, надо хотя бы в общих чертах понимать принципы их работы.

Три этапа работы поискового указателя. Работу поискового указателя можно условно разделить на три этапа. Из них два этапа являются подготовительными — они незаметны для клиента, и лишь на третьем этапе происходит взаимодействие с пользователем, но от каждого из этапов зависят функциональные свойства поисковой системы и эффективность работы с ней.

Сбор первичной базы данных. На первом этапе поисковая система занимается сканированием информационного пространства World Wide Web. Для этого используют специальные агентские программы — *черви* (или пауки). Не следует путать агентов поисковых систем с разновидностью сетевых компьютерных вирусов, тоже именуемых *червями*. Черви поисковых систем совершенно безобидны для серверов и клиентов WWW. По своей сути это очень эффективные малоразмерные браузеры. Им не надо выполнять функции просмотра и воспроизведения содержимого — их задача состоит только в том, чтобы автоматически разыскивать

веб-ресурсы, следуя по гиперссылкам, и, убедившись, что этот ресурс системе еще не известен, копировать его в свою базу данных. Так же происходит и обновление ранее принятых документов, но измененных за время после предыдущего копирования.

Индексация базы данных. Собрать базу данных сетевых веб-ресурсов — еще не значит получить функционирующую поисковую систему. Поиск ключевых слов, введенных пользователем, в обширной базе — это весьма продолжительная операция. Чтобы не задерживать клиента более чем на доли секунды, собранные базы данных проходят предварительную обработку, называемую *индексацией*. На этапе индексации создаются специализированные документы — *поисковые указатели*.

Рафинирование результирующего списка. Это третий этап работы, в ходе которого осуществляется взаимодействие с пользователем. На этом этапе создается список ссылок, который будет передан пользователю в качестве результирующего. Пользовательское представление о качестве работы поисковой системы напрямую зависит от технологий, использованных на этом этапе.

Рафинирование заключается в *фильтрации* и *ранжировании* результатов поиска. Под фильтрацией понимается отсеивание ссылок, которые выдавать пользователю нецелесообразно. Прежде всего проверяется наличие дубликатов. Если система в одном списке выдает множество ссылок, ведущих к одному и тому же веб-ресурсу, это говорит о том, что ее средства добросовестно отработали два первых этапа, но ничего не сделали на третьем этапе. Дублирующиеся ссылки перегружают результирующий список и затрудняют выбор действительно полезных ресурсов.

Ранжирование заключается в создании специального порядка представления результирующего списка, при котором наиболее «полезные» (с точки зрения поисковой системы) ссылки приводятся в вершине списка, а наименее полезные — в его конце. Понимание критерия «полезности» для клиента той или иной ссылки может быть самым разнообразным. Именно поэтому разные поисковые системы, даже работающие с одинаковыми базами ресурсов, выдают разные результаты поиска.

Поисковые системы реального времени. К таким поисковым системам относится, например, alexa.com. Для работы с такой службой пользователь должен подключиться к ее центральному серверу, получить оттуда и установить на своем компьютере клиентскую программу. Эта программа подключается к браузеру и работает как дополнительная панель.

При каждом запуске браузера клиентская программа устанавливает соединение со своим центральным сервером и далее работает с ним в паре. Она передает серверу копии всех веб-страниц, которые посещает пользователь, то есть выполняет те же функции, что и автоматический *червь*, копирующий веб-ресурсы на сервер традиционной поисковой системы. Однако при этом есть два существенных различия:

- во-первых, человек в ходе навигации в WWW руководствуется не теми принципами, что автоматическая программа, поэтому сервер получает копии не всех веб-ресурсов, а только тех, что заинтересовали кого-то из его клиентов;
- во-вторых, если поставкой веб-ресурсов занимаются несколько миллионов постоянных клиентов, индексация веб-пространства происходит намного быстрее.

В свою очередь, пользователь тоже имеет важное преимущество. На какой бы веб-странице он ни находился, система всегда готова предложить ему список других веб-страниц, имеющих близкое по тематике содержание. Она готовит этот список на основании предшествующего опыта, полученного в работе с другими людьми. Так можно получить рекомендации, которые было бы очень трудно (а зачастую и невозможно) разыскать в WWW традиционными поисковыми средствами.

Метапоисковые системы. Такие системы, например nigma.ru, позволяют одновременно использовать множество обычных поисковых систем. Метапоисковая система, в общем случае, не имеет собственных поисковых индексов, собственных поисковых роботов. Она отправляет запрос, введенный пользователем, другим поисковым системам, получает от них результаты поиска, обрабатывает их и выдает пользователю. В некоторых случаях, наряду с использованием других поисковых систем, метапоисковые системы имеют собственные поисковые мощности.

Поиск изображений и видеофайлов. Поиск по медиаконтенту предлагают многие традиционные поисковые системы. Существуют и специализированные поисковые системы, ориентированные на подобный поиск. Например, система picsearch.com. В случае с изображениями и видеофайлами поиск осуществляется по ключевым словам, которые находятся в текстах, находящихся на тех же страницах, что и медиафайлы, а также по названиям файлов.

Специализированные поисковые системы. Специализированные поисковые системы могут быть ориентированы на поиск новостей, работы, научных публикаций. Они предлагают поиск по узкоспециализированным базам данных.

Рекомендации по приемам эффективного поиска

При проведении первичного реферативного поиска, когда тема задана достаточно широко, целесообразно использовать поисковые каталоги. Это позволит быстро установить местоположение основных первоисточников. При ознакомлении с первоисточниками следует, прежде всего, уделять внимание понятийной базе. Знание основных понятий и терминов позволит перейти к углубленному поиску в поисковых указателях с использованием ключевых слов, наиболее точно характеризующих тему.

При наличии первичных сведений по теме поиска документы можно разыскивать в поисковых указателях. При этом следует различать приемы *простого, расширенного, контекстного и специального поиска*.

- ❑ Под *простым поиском* понимается поиск веб-ресурсов по одному или нескольким ключевым словам. Недостаток простого поиска заключается в том, что обычно он выдает слишком много документов, среди которых трудно выбрать наиболее подходящие.
- ❑ При использовании *расширенного поиска* ключевые слова связывают между собой операторами логических отношений. Расширенный поиск применяют в тех случаях, когда приемы простого поиска дают слишком много результатов. С помощью логических отношений поисковое задание формируют так, чтобы более точно детализировать задание и ограничить область отбора, например по дате публикации или по типу данных.

- ❑ *Контекстный поиск* — это поиск по точной фразе. Он удобен для реферативного поиска информации, но доступен далеко не во всех поисковых системах. Прежде всего, чтобы обеспечивать такую возможность, система должна работать не только с индексированными файлами, но и с полноценными образами веб-страниц. Эта операция достаточно медленная, и ее выполняют лишь немногие поисковые системы.
- ❑ *Специальный поиск* применяют при розыске веб-страниц, содержащих ссылки на заданные адреса URL, а также содержащих заданные данные в служебных полях, например в поле заголовка.

Рекомендации по использованию поисковых систем

Самым большим поисковым указателем обладает поисковая система *Google* (www.google.com). Она имеет русскоязычную версию (www.google.ru).

В России в настоящее время наиболее эффективно использовать поисковую систему Яндекс (www.yandex.ru), обеспечивающую максимальный охват российского сектора WWW. Она сочетает в себе возможности поискового каталога и поискового указателя. Особенно удобно использовать ее при формировании сложных поисковых заданий, поскольку она обладает очень гибким языком для расширенного поиска.

Для того чтобы получить быстрый доступ к поисковой системе, можно воспользоваться полем Поиск, которое находится в правом верхнем углу окна Internet Explorer 8.0. Поисковый запрос можно вводить непосредственно в это поле. Результаты поиска будут отображены в обычном окне браузера, в котором будет открыта страница той поисковой системы (по умолчанию это поисковая система Bing от Microsoft), которой был направлен запрос.

9.4. Отправка и получение сообщений

Для работы с электронной почтой и телеконференциями обычно используют единую программу, так как и в том и в другом случае речь идет об отправке и приеме сообщений. Часто оказывается удобным объединение средств работы с этими службами в рамках одной программы. Например, так сделано в программе *Почта Windows Live*, которая позволяет получать и отправлять сообщения электронной почты и телеконференций, используя аналогичные средства.

В Windows 7 нет встроенной почтовой программы. Однако Microsoft предлагает всем желающим скачать и установить бесплатный пакет программ Windows Live, который, кроме прочего, включает в себя почтовый клиент Почта Windows Live. Предшественником этой программы был популярный ранее почтовый клиент Outlook Express, который поставлялся вместе с операционной системой.

Электронная почта является одним из старейших сервисов Интернета, который высоко востребован и в современных условиях. Телеконференции постепенно теряют популярность, хотя они все еще используются.

Возможность использования электронной почты сегодня не рассматривается как самостоятельная услуга и автоматически предоставляется тем, кто подключа-

ется к Интернету без дополнительной оплаты. Адрес электронной почты состоит из двух частей. Доменный адрес условно соответствует двум последним частям обозначения компьютера в адресе URL и фактически представляет собой адрес локальной сети, к которой принадлежит конкретный пользователь. Вторая часть адреса (которая в записи идет перед первой и отделяется от нее символом «@») указывает конкретного пользователя в этой локальной сети. Сообщения для данного адресата накапливаются на *почтовом сервере*, а затем передаются на компьютер адресата по запросу.

Например, пользователь, подключающийся к Интернету через поставщика услуг *ABCDE*, может иметь адрес типа *myname@abcde.ru*.

Почтовый ящик можно совершенно бесплатно зарегистрировать на одном из почтовых сервисов. Например, популярностью пользуются почтовые сервисы *mail.ru*, *yandex.ru*, *rambler.ru*, *gmail.com*.

Телеконференции (или *группы новостей*) представляют собой средства распространения сообщений, не предназначенных для конкретного адресата. Информация о наличии сообщения постепенно распространяется от одного *сервера новостей* к другому. Сообщение хранится на сервере в течение некоторого времени (от нескольких дней до нескольких недель) после чего сбрасывается. Пользователь имеет доступ ко всем сообщениям, имеющимся на данном сервере новостей.

Авторы сообщений направляют их в тематические телеконференции. Имена телеконференций образуют иерархическую структуру, не имеющую единого корня. Элементы имени разделяются точками, старшие элементы располагаются слева, младшие — правее. Чем больше элементов в имени телеконференции, тем более узкой теме она посвящена.

При обращении к телеконференции сервер новостей передает на компьютер пользователя заголовки имеющихся в ней и не прочитанных пользователем сообщений. Текст сообщений передается позже в соответствии с указаниями пользователя и настройками программы чтения сообщений телеконференций. Можно также отправить в телеконференцию новое сообщение или отклик.

Хотя электронная почта и служба новостей — разные службы, для пользователя они почти одинаковы, так как и в том и в другом случае речь идет об отправке и получении сообщений.

Сообщение, отправляемое в телеконференцию, носит общественный характер, а частную информацию следует пересылать по электронной почте. Однако ни одна из этих служб не годится для пересылки *конфиденциальной* информации, которая не должна быть доступна посторонним.

Веб-каналы. Другое название этой технологии звучит как *RSS-каналы*. Подписавшись на веб-канал, вы сможете получать актуальную информацию с сайта, которому принадлежит канал. Например, новостные сайты рассылают с помощью этой технологии краткие анонсы свежих новостей.

Работа с программой Почта Windows Live

Создание учетной записи

Сообщения электронной почты и телеконференций накапливаются соответственно на *почтовом сервере* и *сервере новостей*. Для работы с этими службами предназна-

чена программа Почта Windows Live (Пуск ► Все программы ► Windows Live ► Почта Winows Live) (рис. 9.8).

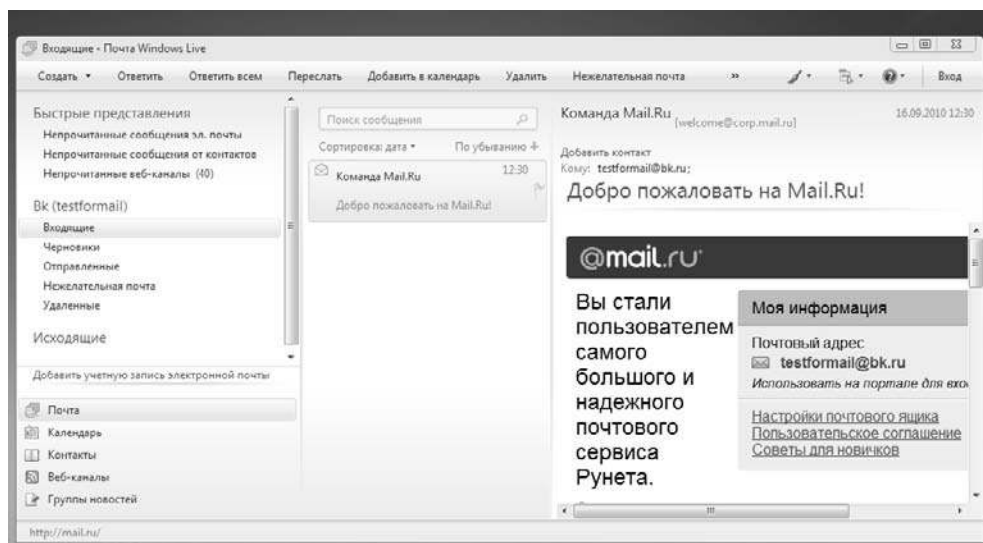


Рис. 9.8. Окно программы Почта Windows Live

Окно программы разделено на несколько областей. Слева находится панель навигации. С ее помощью можно перемещаться между папками, в которых хранятся почтовые сообщения, между режимами работы с почтой, календарем, контактами, веб-каналами и группами новостей. Центральную часть окна занимает область просмотра папок. Здесь, например, может отображаться список писем, новостей, пришедших по RSS-каналу. Правая часть окна отведена под детальный просмотр выбранного сообщения. Верхнюю часть окна занимает панель инструментов.

Так как сообщения поступают и отправляются через сервер, программе требуется указать информацию об используемом сервере. Эта информация хранится в виде *учетной записи*.

Прежде чем запускать программу, следует подготовить данные, необходимые для настройки доступа к почтовому ящику. Обычно это следующая информация:

- ☐ адрес электронной почты (например, testformail@bk.ru);
- ☐ пароль доступа к электронной почте;
- ☐ адрес SMTP-сервера (например, smtp.bk.ru);
- ☐ адрес POP-сервера (например, pop.bk.ru).

Если почтовый ящик предоставляет провайдер интернет-услуг, эти сведения обычно прилагаются к договору о подключении к Интернету. Если вы зарегистрировали почтовый ящик на одном из онлайн-почтовых сервисов, необходимые данные можно найти в справочном разделе сервера, посвященном настройке почтовых клиентов.

При первом запуске программа предложит создать новую учетную запись электронной почты. Также добавить в программу новую учетную запись (она под-

держивает работу с несколькими учетными записями) можно, воспользовавшись ссылкой **Добавить учетную запись электронной почты** на навигационной панели окна программы.

Последующая информация вводится под управлением мастера и включает имя, указываемое как имя отправителя, адрес электронной почты, имя используемого сервера и, в случае необходимости, имя пользователя и пароль.

Для управления учетными записями, в частности для их редактирования, нам понадобится воспользоваться классическим меню программы, которое по умолчанию скрыто. Для отображения меню нужно нажать клавишу **Alt** или **F10** на клавиатуре, после чего выбрать команду **Сервис ► Учетные записи**.

Создание сообщения электронной почты

Чтобы отправить сообщение электронной почты, его надо создать. Для этого следует щелкнуть на кнопке **Создать** на панели инструментов. Если воспользоваться меню кнопки, то для создания сообщения электронной почты необходимо выбрать пункт **Сообщение электронной почты**.

При этом открывается окно **Создать сообщение** (рис. 9.9), рабочая область которого разбивается на две основные части. В верхней части располагаются поля для ввода служебной информации, а в нижней — собственно текст сообщения. В поле **Тема** вводится краткое описание вопроса, которому посвящено сообщение. После того как тема указана, соответствующий текст становится заголовком окна. В поле **Кому** вводится адрес основного получателя письма.

Если необходимо отправить копию письма другому получателю или отправить копию письма, о которой ничего не известно другим адресатам, соответствующие адреса вводятся в поля **Копия** и **Скрытая копия**. Если эти поля не видны в окне, их нужно отобразить, нажав на ссылку **Отображать**.

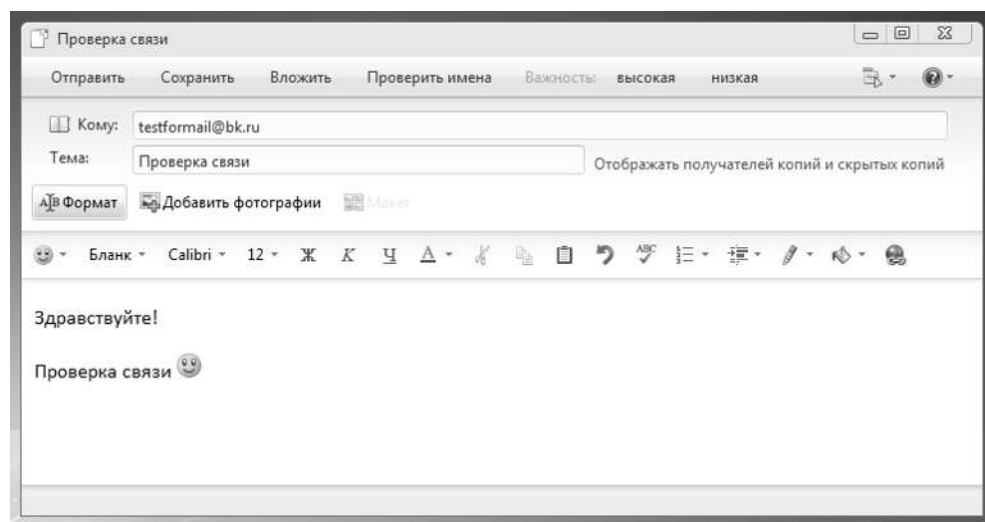


Рис. 9.9. Создание сообщения для отправки по электронной почте

В ходе создания и редактирования сообщения наличие связи с почтовым сервером не требуется. Такая связь нужна только в момент отправки (получения) сообщений. Программа Почта Windows Live устроена таким образом, что отправка и получение сообщений осуществляются одновременно. Так, получение и доставка почты осуществляются по щелчку на кнопке **Отправить** в окне создания сообщения или по щелчку на кнопке **Синхронизировать** (с выбором пункта меню, соответствующего синхронизации сообщений для выбранной учетной записи или для всех учетных записей) в основном окне программы Почта Windows Live.

Сообщения электронной почты размещаются в системе «внутренних» папок программы. Поступившие сообщения заносятся в папку **Входящие**. Открыв эту папку щелчком на ее значке на панели навигации, можно увидеть в правой области список поступивших сообщений. Если выбрать щелчком любое из сообщений, его содержание отобразится в области, расположенной правее списка. Двойной щелчок позволяет открыть и прочитать сообщение в отдельном окне.

Подготовка ответов на сообщения

Как правило, использование любых средств коммуникации подразумевает диалог. В случае электронной почты речь идет об отправке ответов на полученные сообщения. Программа Почта Windows Live включает средства, упрощающие подготовку таких ответов. Открыв полученное сообщение в отдельном окне, можно использовать кнопки на панели инструментов. Также можно воспользоваться контекстным меню письма в списке или, при выделении письма, кнопками на панели инструментов основной программы.

- ❑ Кнопка **Ответить** служит для ответа автору письма. При этом в окне создания сообщения автоматически заполняются поля **Кому** и **Тема**, а в «тело» сообщения заносится текст исходного сообщения, что позволяет привязать комментарии непосредственно к отдельным фразам полученного письма.
- ❑ Кнопка **Ответить всем** служит для отправки ответа автору письма, а также всем, кто получил исходное сообщение. В окне создания сообщения автоматически заполняются поля **Кому**, **Копия** и **Тема**. Текст исходного сообщения копируется в тело сообщения.
- ❑ Кнопка **Переслать** позволяет отправить полученное сообщение (вместе с комментариями, если необходимо) другому корреспонденту. В данном случае автоматически заполняется только поле **Тема**, так как нового адресата необходимо указать дополнительно.

Чтение сообщений телеконференций

Механизм чтения сообщений телеконференций примерно тот же, что и при использовании электронной почты. На панели навигации находится значок, для доступа к сообщениям телеконференции. После выбора этого значка можно воспользоваться ссылкой **Добавить учетную запись группы новостей**. Выбрав телеконференцию, следует щелкнуть на кнопке **Подписаться**.

Работа с сообщениями телеконференций осуществляется примерно так же, как с сообщениями электронной почты

Чтение веб-каналов

Для чтения веб-канала нужно предварительно узнать адрес этого канала (например, <http://lenta.ru/rss/>), после чего перейти к работе с веб-каналами, используя панель навигации. Сообщения веб-канала можно читать так же, как обычные письма, пересылать их. Отвечать на сообщения веб-канала нельзя. Он подразумевает одностороннюю коммуникацию.

Работа с контактами

При активном использовании электронной почты общее число корреспондентов может достигать многих сотен. Помнить все электронные адреса просто невыносимо. Облегчить эту работу позволяет инструмент Контакты Windows Live, встроенный в Почту Windows Live.

С помощью этого инструмента можно:

- ☐ запоминать адреса корреспондентов, от которых поступили сообщения;
- ☐ автоматизировать ввод адресов корреспондентов;
- ☐ организовать проверку правильности введенных адресов;
- ☐ упростить отправку сообщений группам адресатов.

Для того чтобы добавить адресата в Контакты, достаточно воспользоваться соответствующей командой контекстного меню письма или, открыв письмо, нажать на ссылку **Добавить контакт**, которая расположена около адреса отправителя письма.

Добавлять адреса можно непосредственно в окне работы с Kontakтами (кнопка Контакты на панели навигации) (рис. 9.10).

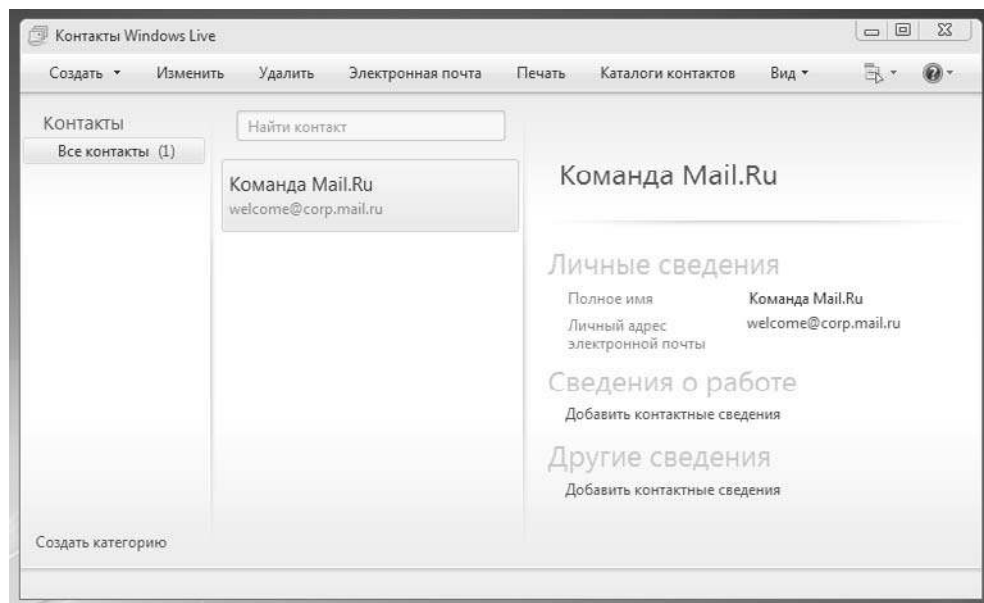


Рис. 9.10. Контакты Windows Live

Чтобы воспользоваться Контактами Windows Live для ввода адреса, надо в ходе создания сообщения щелкнуть на заголовке соответствующего поля (Кому, Копия или Скрытая копия). Адреса, помещаемые в каждое из этих полей, выбираются в диалоговом окне выбора получателей.

Адрес, взятый из Контактов, выделяется в соответствующем поле подчеркиванием. Если какие-то из адресов вводились вручную, но должны быть в Контактах, их можно проверить при помощи команды Сервис ► Проверить имена. Найденные адреса также будут подчеркнуты, ненайденные можно исправить, выбрав один из нескольких подходящих адресов, или занести в Контакты Windows Live.

Если необходимо регулярно отправлять сообщение одной и той же группе корреспондентов, Адресная книга позволяет создать и использовать *группу адресов*. Для этого используется команда Создать ► Категорию. При добавлении участников в группу их адреса могут выбираться из Контактов Windows Live или создаваться на месте. При указании в поле адреса имени группы сообщение отправляется всем выбранным корреспондентам.

Включение корреспондента в группу не влияет на возможность индивидуального использования его адреса. Один корреспондент может быть включен в несколько групп.

Практическое занятие

Упражнение 9.1. Поиск информации по ключевым словам

1. Запустите Internet Explorer (Пуск ► Все программы ► Internet Explorer).
2. На панели Адрес введите: <http://www.yandex.ru/> и щелкните на кнопке Переход или нажмите клавишу ENTER на клавиатуре.
3. Внимательно рассмотрите загруженную страницу, найдите поле для ввода ключевых слов и кнопку запуска поиска. Мы собираемся искать веб-страницы, посвященные простым механизмам.
4. В поле для ввода ключевых слов введите простые механизмы.
5. Щелкните на кнопке Найти.
6. Просмотрите результаты поиска.
7. Щелкните на гиперссылке с номером 1.
8. Просмотрите загруженную страницу.
9. Поисковая система Яндекс всегда открывает найденную страницу в отдельном окне. Закройте это окно и вернитесь к результатам поиска.
10. Повторяя действия пп. 7–9, просмотрите всю первую группу из десяти ссылок на найденные страницы. Для удобства вы можете открывать ссылки не в новом окне, а в новой вкладке, воспользовавшись соответствующей командой контекстного меню ссылки. Сколько из этих страниц все еще существуют? Сколько из них можно считать полезными?
11. Щелкните на поле ввода Поиск на панели инструментов.
12. Введите набор ключевых слов из п. 4 в поле Поиск.

13. Щелкните на кнопке начала поиска.
14. Сравните результаты поиска.
15. На панель Адрес введите слово find и набор ключевых слов из п. 4. Щелкните на кнопке Переход.
16. Объясните, что произошло.

Итог. Мы научились проводить поиск информации в Интернете тремя разными способами: с помощью поисковой системы, с помощью панели Поиск и непосредственно с панели Адрес. Мы узнали, в чем состоят особенности поиска по ключевым словам.

Упражнение 9.2. Использование папки Избранное

1. Запустите Internet Explorer.
2. На панели Адрес введите: <http://top100.rambler.ru/navi/?theme=708/709&src=cat> (или другой адрес по указанию преподавателя) и щелкните на кнопке Переход.
3. Просмотрите загруженную страницу.
4. Щелкните в рабочей области программы правой кнопкой мыши и выберите в контекстном меню команду Добавить в папку «Избранное».
5. В поле Имя введите: Экспериментальная страница.
6. Щелкните на кнопке ОК.
7. Щелкните на кнопке Домой на панели инструментов.
8. Откройте панель Избранное кнопкой Избранное на панели инструментов.
9. Убедитесь, что в папке Избранное действительно была сохранена информация о загружаемой странице.
10. Дайте команду Упорядочить избранное из меню панели Избранное. Щелкните на кнопке Создать папку. Дайте новой папке имя Компьютеры.
11. Выберите пункт Экспериментальная страница. Щелкните на кнопке Переместить.
12. В диалоговом окне Обзор папок выберите папку Компьютеры, после чего щелкните на кнопке ОК.
13. Закройте диалоговое окно Упорядочить избранное.
14. Найдите папку Компьютеры на панели избранного, перейдите в нее, откройте помещенную в нее страницу.
15. Ознакомьтесь с тем, какая страница при этом загружается.
16. Уничтожьте папку Компьютеры и все ее содержимое.

Итог. Мы научились сохранять информацию о полезных веб-страницах в папке Избранное. Мы также узнали, как изменять структуру папок, вложенных в папку Избранное, и познакомились с загрузкой избранных веб-страниц.

Упражнение 9.3. Загрузка файла из Интернета

1. Запустите Internet Explorer.
2. На панели Адрес введите: <ftp://ftp.microsoft.com/> и щелкните на кнопке Переход.
3. Внимательно рассмотрите способ представления каталога архива FTP в программе Internet Explorer.
4. Щелчками по ссылкам папок откройте папку /Products/Windows/Windows95/CDRomExtras/FunStuff/.
5. Щелкните по ссылке clouds.exe. Она ведет к сжатому файлу, содержащему графический файл фона рабочего стола, который применялся в операционной системе Windows 95.
6. В открывшемся диалоговом окне Загрузка файла щелкните на кнопке Сохранить.
7. В диалоговом окне Сохранить как выберите папку, специально отведенную для хранения загруженных файлов, и задайте имя файла.
8. Сбросьте в диалоговом окне загрузки файла флажок Закрывать диалоговое окно после завершения загрузки.
9. Следите за ходом загрузки файла по этому диалоговому окну.
10. Когда загрузка файла завершится, закройте диалоговое окно, информирующее о завершении загрузки, с помощью кнопки Закрывать.
11. Откройте папку, в которой был сохранен загруженный файл, при помощи программы Проводник.
12. Убедитесь, что загруженный файл можно использовать в соответствии с его назначением.

Итог. Мы научились просматривать каталоги FTP и загружать файлы из Интернета. Механизм загрузки файлов работает практически одинаково при загрузке с веб-узла и из архива FTP.

Упражнение 9.4. Настройка отображения объектов

1. Запустите Internet Explorer.
2. На панели Адрес введите: <http://elfwood.com> (или другой адрес по указанию преподавателя).
3. Щелкните на гиперссылке Click here to join...
4. Зафиксируйте с помощью секундомера время загрузки страницы.
5. Посмотрите, как выглядит загруженная страница.
6. Щелкните на кнопке Назад на панели инструментов.
7. Дайте команду Сервис ► Свойства обозревателя.
8. Откройте вкладку Дополнительно.

9. Сбросьте флажки Воспроизводить анимацию на веб-страницах, Воспроизводить звуки на веб-страницах, Показывать изображения.
10. Выберите вкладку Общие.
11. Щелкните на кнопке Удалить.
12. В появившемся окне Удаление истории обзора нажмите на кнопку Удалить.
13. Щелкните на кнопке ОК.
14. Закройте окно Internet Explorer, откройте снова и выполните действия, перечисленные в пп. 2–5.
15. Сравните результаты измерений.
16. Сравните внешний вид страницы при предыдущей и нынешней загрузке.
17. Щелкните на одной из пустых рамок для рисунков правой кнопкой мыши и выберите в контекстном меню команду Показать изображение.

Итог. Мы научились ускорять загрузку веб-страниц ценой отказа от отображения рисунков и других объектов. Мы узнали, как индивидуально загружать нужные объекты. В ходе упражнения мы также выяснили, как очистить пространство на диске, занятое временными файлами Интернета.

Упражнение 9.5. Создание учетной записи электронной почты

1. Запустите программу Почта Windows Live.
2. Щелкните по ссылке Добавить учетную запись электронной почты.
3. Введите в поле Адрес электронной почты заданный адрес электронной почты.
4. Введите в поле Пароль заданный пароль доступа к почтовому ящику.
5. В поле Отображаемое имя введите свои имя и фамилию.
6. Щелкните на кнопке Далее.
7. Введите заданные имена серверов для входящей и исходящей почты. Если используется один сервер, введите одно и то же имя в оба поля.
8. Щелкните на кнопке Далее.
9. Щелкните на кнопке Готово.
10. Убедитесь в том, что на панели навигации программы появилась информация о новой учетной записи электронной почты.

Итог. Мы научились создавать учетную запись для электронной почты, используемую при отправке и получении корреспонденции. Мы узнали, какие данные потребуются для создания учетной записи.

Упражнение 9.6. Отправка и получение сообщения электронной почты

1. Запустите программу Почта Windows Live.
2. На панели навигации выберите учетную запись, созданную в предыдущем упражнении.

3. Щелкните по кнопке **Создать** на панели инструментов.
4. В поле **Тема** введите слова: **Проверка работы электронной почты**.
5. В поле **Кому** введите заданный (свой собственный) адрес электронной почты.
6. В тело сообщения введите произвольный легко запоминающийся текст.
7. Щелкните на кнопке **Отправить** на панели инструментов.
8. Щелкните на кнопке **Синхронизировать** на панели инструментов.
9. Проследите за процессом отправки созданного сообщения и поиском на сервере поступивших сообщений.
10. Убедитесь, что только что отправленное сообщение появилось в списке поступивших сообщений.
11. Выберите это сообщение в списке и ознакомьтесь с его содержанием на правой панели.
12. Дважды щелкните на заголовке сообщения, чтобы открыть его в отдельном окне.
13. Закройте окно сообщения.

Итог. Мы научились создавать сообщения электронной почты, отправлять их, а также получать и читать поступившие сообщения. Мы также познакомились с тем, как организуется процесс передачи сообщений с компьютера пользователя на почтовый сервер и в обратную сторону.

Упражнение 9.7. Подписка на веб-канал

1. Запустите программу **Почта Windows Live**.
2. Щелкните на ссылке навигационной панели **Веб-каналы**.
3. В окне управления веб-каналами нажмите на ссылку **Добавить веб-канал**.
4. В поле **URL-адрес** введите: <http://lenta.ru/rss/>.
5. Щелкните на кнопке **ОК**. Если компьютер, на котором вы выполняете упражнение, подключен к Интернету, начнется загрузка сообщений.
6. Щелкните на ссылке выбранного канала на панели навигации.
7. Выберите какое-либо сообщение, чтобы просмотреть его. Щелкните на сообщении дважды, чтобы открыть его в отдельном окне.
8. Закройте окно сообщения.

Итог. Мы научились производить подписку на веб-каналы (RSS-ленты). Мы также узнали, как получать сообщения с сервера и читать имеющиеся сообщения.

Упражнение 9.8. Использование Контактов Windows Live

1. Запустите программу **Почта Windows Live**.
2. Создайте вручную запись в **Контактах Windows Live** для своего адреса электронной почты. Для этого нажмите кнопку **Контакт** на панели инструментов.

3. Заполните поля **Фамилия** и **Имя** на вкладке **Основные**.
4. Укажите адрес электронной почты и щелкните на кнопке **Добавить контакт**.
5. По желанию занесите данные о себе также на вкладки **Контактные**, **Личные**, **Рабочие** и на другие вкладки.
6. Щелкните на кнопке **Добавить контакт**.
7. Убедитесь, что данные занесены в **Контакты Windows Live**, — на панели **Контакты** должен добавиться новый значок.
8. Активируйте окно **Почта Windows Live**, перейдите в папку **Входящие** вашей учетной записи.
9. Щелкните на заголовке входящего сообщения правой кнопкой мыши и выберите в контекстном меню пункт **Добавить отправителя в список контактов**.
10. Убедитесь, что данные занесены в **Контакты Windows Live**.
11. Активируйте окно **Почта Windows Live**, щелкните на кнопке **Создать** на панели инструментов.
12. Щелкните на кнопке **Кому** рядом с полем ввода адреса.
13. Выберите собственный адрес в списке слева и щелкните на кнопке **Кому**.
14. Щелкните на кнопке **ОК**.
15. Убедитесь, что имя адресата внесено в поле **Кому**. Обратите внимание, что использовано именно имя, а не адрес. Обратите внимание, что имя подчеркнуто. Это означает, что данный адрес считается «проверенным».
16. Произвольным образом заполните поле **Тема**, а также введите текст сообщения.
17. Отправьте созданное сообщение и убедитесь, что оно доставляется правильно (см. упражнение 9.6).

Итог. Мы научились заносить адреса электронной почты в **Контакты**, вручную и извлекая их из поступивших сообщений. Мы также узнали, как использовать данные из **Контактов** при отправке сообщений, и выяснили, что **Контакты** позволяют использовать имена корреспондентов вместо адресов электронной почты.

Глава десятая

Создание простых текстовых документов

В этой и следующей главе рассматриваются понятия, методы и приемы, относящиеся к созданию текстовых документов с помощью персонального компьютера. Условно (из чисто методических соображений) мы выделим две группы создаваемых документов — *простые* и *комплексные*. Первые представляют собой форматированный текст, а вторые содержат кроме текста объекты иной природы (чертежи, рисунки, формулы, таблицы, объекты мультимедиа и пр.).

10.1. Общие сведения о текстовом процессоре Microsoft Word

Общее название программных средств, предназначенных для создания, редактирования и форматирования простых и комплексных текстовых документов, — *текстовые процессоры*. В настоящее время в России наибольшее распространение имеет текстовый процессор *Microsoft Word*. Это связано, прежде всего, с тем, что его создатели относительно давно предусмотрели *локализацию* программы в России путем включения в нее средств поддержки работы с документами, выполненными на русском языке.

Сегодня вы можете столкнуться с различными версиями Microsoft Word.

Рабочее окно процессора Microsoft Word 2010

Рабочее окно процессора Microsoft Word 2010 представлено на рис. 10.1. Его основные элементы управления: панель быстрого запуска, лента, на которой расположены вкладки (аналог пунктов меню в прежних версиях программы), содержащие панели инструментов, рабочее поле и строка состояния, включающая индикаторы и элементы управления.

Режимы отображения документов

Текстовый процессор Microsoft Word поддерживает несколько режимов представления документов.

В *режиме черновика* представляется только содержательная часть документа без реквизитных элементов оформления, относящихся не к тексту, а к печатным страницам (колоннатулы, колонцифры, подстраничные сноски и т. п.). Этот режим удобен на ранних этапах разработки документа (ввод текста, редактирование, рецензирование), а также во всех случаях, когда содержательная часть документа имеет более высокое значение, чем внешнее представление. В этом режиме операции с объемными документами проходят быстрее, что важно при работе на малопроизводительных компьютерах.

В режиме веб-документа экранное представление не совпадает с печатным. Это отступление от принципа *WYSIWYG*, но оно характерно для электронных публикаций в World Wide Web, поскольку заранее не известно, каким средством просмотра и на каком оборудовании будет отображаться документ. Понятие печатной страницы для электронных документов не имеет смысла, поэтому назначенные параметры страницы не учитываются, а форматирование документа на экране является *относительным*. В этом режиме разрабатывают электронные публикации.

В *режиме разметки* экранное представление документа полностью соответствует печатному, вплоть до назначенных параметров печатной страницы. Этот режим удобен для большинства работ, связанных с форматированием текста, предназначенного для печати.

Режим структуры предназначен для работы со структурой документа. В этом режиме отображаются заголовки документа, которые формируют его структуру. Также в этом режиме можно просматривать содержательную часть документов, однако работа с ней не играет определяющей роли. Режим полезен в тех случаях, когда разработку документа начинают с создания плана содержания. Если предполагаемый размер документа превышает 5–7 печатных страниц, следует начинать работу именно с создания первичного плана. Режим структуры отличается тем, что при его включении автоматически открывается вспомогательная вкладка Структура, элементы управления которой позволяют править структуру документа.

Режим чтения предназначен для чтения и комментирования документов. В этом режиме документ выводится в максимально удобном для чтения полноэкранном виде. Редактировать документ в режиме чтения нельзя, однако можно добавлять примечания и выделять цветом некоторые участки текста.

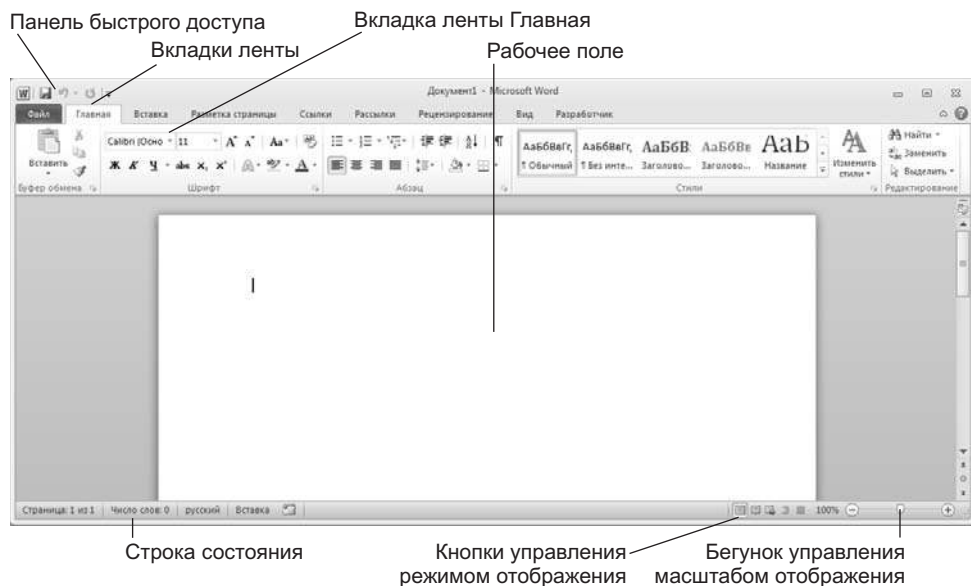


Рис. 10.1. Рабочее окно программы Word 2010

Выбор одного из пяти указанных режимов представления документа выполняются с помощью группы параметров **Режимы просмотра документов** вкладки **Вид** или с помощью командных кнопок, расположенных в правом нижнем углу окна приложения.

При необходимости в любом режиме представления документа можно подключить окно навигации, которое выводит структуру документа. Это окно полезно при навигации по объемному документу. Подключение окна навигации производится установкой флажка **Область навигации** в группе параметров **Показать вкладки Вид**.

Через меню **Файл** доступен один режим представления документа, используемый для предварительного просмотра документа перед печатью. Его можно вызвать, выполнив команду **Файл ► Печать**. В правой части окна настроек печати можно будет увидеть документ в таком виде, в котором он будет выведен на печать.

Приемы работы с командами ленты

Ленточный интерфейс, впервые появившийся в Microsoft Office 2007, пришел на смену интерфейсу, основным элементом которого было обычное меню. Такой переход не случаен. В частности, ленточный интерфейс позволяет нормально работать с программой, используя весьма широкий диапазон экранных разрешений. Учитывая современное состояние компьютерной техники, когда популярностью пользуются различные мобильные устройства с небольшими разрешениями экранов (ноутбуки, субноутбуки, планшетные компьютеры), переход на ленточный интерфейс оправдан.

Лента содержит вкладки, элементы управления вкладок разбиты на группы параметров. В каждой из групп представлены наиболее важные и часто используемые элементы управления. При необходимости из большинства групп параметров можно вызвать диалоговые окна и панели инструментов, содержащие дополнительные параметры.

Лента Microsoft Word 2010

Программа Microsoft Word поддерживает возможность самостоятельной настройки вкладок ленты. Настройку выполняет пользователь путем подключения и отключения необходимых ему по роду деятельности элементов вкладок (**Файл ► Параметры ► Настройка ленты**) (рис. 10.2).

Некоторые вкладки из перечисленных в окне настройки по умолчанию скрыты. Например, вкладка **Разработчик** нужна для тех, кто занимается разработкой программ на VBA для Microsoft Office. Для того чтобы отобразить эту вкладку в общем списке, нужно установить флажок около ее названия.

Некоторые вкладки отображаются на экране лишь тогда, когда в них есть необходимость, например при выделении рисунка на экране появится вкладка **Работа с рисунками ► Формат**, при работе с таблицами вы увидите группу элементов **Работа с таблицами**, которая включает в себя вкладки **Конструктор** и **Макет**.

Вкладка ленты	Состав, назначение элементов управления	Примечание
Вид	Управление внешним видом документа при редактировании, переключение между режимами просмотра документа, управление масштабом документа	
Разработчик	Создание и редактирование макросов, настройка защиты документа, работа с XML и элементами управления, которые позволяют организовать интерактивное использование электронного документа	Создание макросов возможно как в автоматическом (посредством записи действий пользователя), так и в ручном режиме с помощью редактора Visual Basic, встроенного в продукты Microsoft Office

Кроме того, в программе *Word 2010* роль специальной контекстно-зависимой информационно-инструментальной панели играет область задач. Эта область открывается на правах панели инструментов и обычно располагается у правого края окна программы. Область задач может использоваться для выполнения разных функций, в зависимости от источника вызова. Например, при попытке вставить в документ картинку (Вставка ► Иллюстрации ► Картинка) у правого края окна появится панель задач Картинка, которая позволяет искать картинки.

Основные принципы практической работы с текстовым процессором Microsoft Word

Основные принципы практической работы зависят от используемой версии программы. Базовый принцип здесь состоит в том, что чем больше возможностей имеет программа, тем строже надо подходить к выбору тех функций, которыми можно пользоваться в каждом конкретном случае. Удобен подход, когда набор допустимых средств оформления и форматирования документа определяет его заказчик.

Заказчик есть у каждого документа. Даже если документ готовится для личного употребления, условным заказчиком является сам автор. Заказчиком можно считать и исполнителя, которому передается документ для последующих операций, например для рецензирования или вывода на печать. К категории «заказчиков» относятся и предполагаемые клиенты, для которых данный документ разрабатывается. При этом возникает ряд вопросов, которые надо решить до начала работы с документом.

К какому типу относится документ? Современные текстовые процессоры позволяют создавать документы трех типов. Во-первых, это *печатные документы*, которые создаются и распечатываются на одном рабочем месте или в одной рабочей группе. Дальнейшее движение документа происходит только в бумажной форме. Состав допустимых средств оформления в данном случае определяется только техническими возможностями печатающего устройства.

Второй тип — *электронные документы в формате текстового процессора*, например Microsoft Word. Такие документы передаются заказчику в виде файлов. Электронный документ, как правило, не является окончательным. В большинстве случаев заказчик может его дорабатывать, редактировать, форматировать, распеча-

тивать или использовать его компоненты для подготовки своих документов (книг, журналов, сборников статей и т. п.). Набор разрешенных средств в данном случае, как правило, минимален и определяется заказчиком.

Третий тип — *веб-документы*. Предполагается, что в этом качестве они останутся навсегда и их преобразование в печатные документы не планируется. В веб-документах большую роль играет управление цветом. Для этой категории документов наиболее широк выбор средств форматирования и оформления.

Кто является заказчиком документа? Самый типичный случай — когда заказчиком документа является работодатель, то есть администрация предприятия или учреждения. Надо выяснить правила оформления документов, принятые в данной организации, и строго их придерживаться. Если существуют готовые шаблоны, их надо использовать, а если их нет, то разработать свои и согласовать с руководством.

Самый простой случай — когда заказчика нет, и автор делает документ для себя. Он может использовать любые средства, которые ему подскажет фантазия и которые поддерживаются его устройствами вывода (экран для веб-документов или принтер для печатных документов).

Самый трудный случай — когда заказчик внешний, особенно если он не вполне определен. В этом случае исполнители часто путают понятия *представление* документа и *предоставление* документа. Для *представления* документа они стремятся использовать все средства форматирования, которые наилучшим образом подчеркивают достоинства документа. При *предоставлении* документа ситуация обратная. Здесь не автор, а заказчик определяет форму и средства форматирования и оформления. Использование этих средств в данном случае имеет *разрешительный характер*. Если же требования заказчика еще не известны, следует предполагать, что нежелательны большинство средств форматирования документов, передаваемых для дальнейшей обработки. В частности, необходимо:

- ☐ ограничить используемые наборы шрифтов только теми, которые входят в состав операционной системы (не более двух наборов: один — для основного текста, другой — для заголовков и вспомогательного текста);
- ☐ минимизировать использование средств форматирования абзацев: отказаться от выравнивания по ширине и от переноса слов, ограничить число используемых шрифтовых начертаний (не более двух: основного и дополнительного);
- ☐ отключить все автоматические средства форматирования: расстановку колонтитулов, нумерацию страниц, маркировку и нумерацию списков и пр.;
- ☐ не использовать встроенные средства текстового процессора для создания встроенных объектов (художественные заголовки, векторные рисунки, рамки и пр.) — все объекты должны создаваться специальными программами, храниться в отдельных файлах, вставляться в текст документа методом связывания и прилагаться к файлу документа;
- ☐ исключить использование приемов взаимодействия встроенных объектов с текстом;
- ☐ сохранять готовые документы в простейших форматах, несущих минимум информации о форматировании (для документов Microsoft Word таковыми

являются форматы Обычный текст (.TXT), Текст в формате RTF (.RTF)). Если вы не знаете точно, какой формат документа принимает заказчик, вы можете использовать Word 97 – 2003 (.DOC). Заказчик практически гарантированно сможет открыть файл этого формата. Файлы нового формата (с расширениями .DOCX и .DOCM) следует передавать заказчику лишь в том случае, если имеется уверенность, что он использует Microsoft Word 2010 или Microsoft Word 2007;

- ☐ в каждом случае отступления от этих правил, например при необходимости использовать формулы, таблицы и специальные символы, согласовывать свои действия с заказчиком.

Эти требования к документам, предоставляемым для дальнейшей технологической обработки, связаны с тем, что большинство средств оформления и форматирования текстового процессора являются «вещью в себе». Достоинства этих средств проявляются только при выводе окончательного документа средствами того же самого процессора, будь то вывод на печать, просмотр на экране или публикация в веб-структуре. При обработке данных, содержащихся в документе, другими программными средствами преимущества форматирования и оформления могут оборачиваться тяжкими проблемами.

Первичная настройка текстового процессора Microsoft Word

Приступая к первому знакомству с текстовым процессором *Microsoft Word*, следует выполнить ряд первичных настроек. Некоторые средства автоматизации, имеющиеся в программе, могут отвлекать начинающего пользователя от главной задачи — освоения основных приемов. В ряде случаев из-за работы автоматических средств результаты операций получаются неожиданными — это препятствует установлению обратной связи и эффективному усвоению практических приемов.

Комплекс настроек, рекомендуемых перед началом освоения текстового процессора, приведен в упражнении 10.1.

10.2. Приемы работы с текстами в процессоре Microsoft Word

К базовым приемам работы с текстами в текстовом процессоре Microsoft Word относятся следующие:

- ☐ создание документа;
- ☐ ввод текста;
- ☐ редактирование текста;
- ☐ рецензирование текста;
- ☐ форматирование текста;
- ☐ сохранение документа;
- ☐ печать документа.

Создание документа

В текстовом процессоре Word 2010 принято использовать два метода создания нового документа: на основе готового шаблона или на основе существующего документа. Второй метод проще, но первый методически более корректен.

Создание документа на основе имеющегося документа. Этот метод потенциально опасен, и потому его использование категорически не рекомендуется! Тем не менее им очень широко пользуются, и мы его рассматриваем, чтобы предупредить о возможных опасностях и обратить внимание на правильный порядок действий.

При создании документа на основе существующего документа:

- ☐ открывают готовый документ (Файл ► Открыть);
- ☐ сохраняют его под новым именем (Файл ► Сохранить как);
- ☐ выделяют в нем все содержимое (Главная ► Редактирование ► Выделить ► Выделить все);
- ☐ удаляют его нажатием клавиши DELETE;
- ☐ в результате получают пустой документ, имеющий собственное имя и сохраняющий все настройки, ранее принятые для исходного документа.

Этот метод характерен для начинающих пользователей, не умеющих создавать шаблоны и пользоваться ими. Получив задание у руководителя, они запрашивают образец и приступают к его правке. Метод интуитивно прост, но чреват весьма неприятными ошибками. Если забыть сохранить новый файл под другим именем, можно легко уничтожить ценный документ, даже не успев создать новый. Кроме того, при небрежной правке содержание документа, взятого за основу, может переходить в новый документ. Для рабочих мест, на которых создаются десятки документов в сутки, этот метод весьма опасен.

Создание документа на основе шаблона. Шаблоны — это те же образцы документов, но защищенные от досадных неприятностей. Создание документа на основе готового шаблона выполняется следующим образом.

1. Команда Файл ► Создать открывает список шаблонов, на основе которых можно создать новый документ. Если никаких предпочтений нет, следует выбрать шаблон Новый документ. Созданный документ приобретает имя Документ1, принятое по умолчанию. Его целесообразно сразу же сохранить под «правильным» именем, выбрав для него соответствующую папку и дав команду Файл ► Сохранить.
2. Диалоговое окно Сохранение документа в текстовом процессоре Microsoft Word 2010, представленное на рис. 10.3, практически не отличается от аналогичного окна ранее рассмотренных нами стандартных приложений. Оно обычно предполагает сохранение файла в библиотеку \Документы, но обеспечивает быстрый доступ и к некоторым иным папкам.
3. В левой части окна Сохранение имеются ссылки, позволяющие быстро выбрать место для сохранения файла.

Документы — это библиотека, которая включает в себя несколько папок, используемых для хранения пользовательских файлов. Она является преемником

традиционной папки Мои документы, которая использовалась для хранения авторских документов в операционных системах семейства Windows предыдущих поколений.

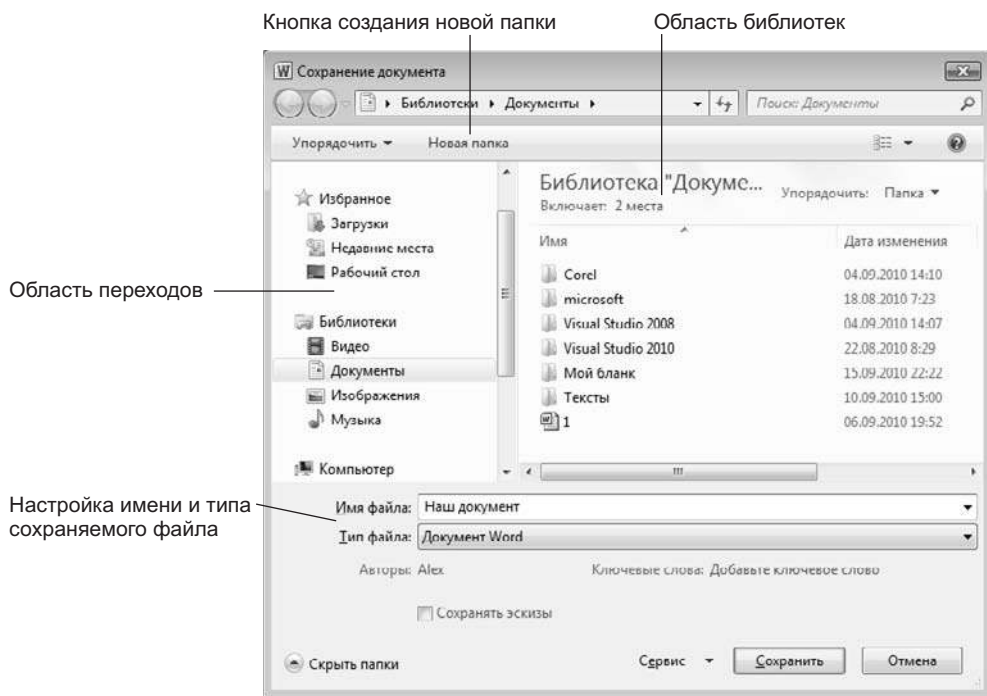


Рис. 10.3. Диалоговое окно Сохранение документа

Рабочий стол — не слишком удобное место для хранения документов, поскольку его принято содержать «в чистоте». Есть два случая, когда Рабочий стол используют для хранения документов:

- если документ временный и после просмотра будет удален в Корзину;
- если документом предполагается пользоваться особенно часто (например, это список номеров телефонов коллег по работе).
- Компьютер — позволяет сохранить файл в произвольную папку.
- Сеть — этот значок обеспечивает быстрый доступ к сохранению документа не на своем компьютере, а в локальной сети, например на файловом сервере. При этом требуются дополнительные операции по навигации, связанные с выбором конкретной папки на конкретном сетевом компьютере.

Специальные средства ввода текста

Технология ввода текста и переключения языковых раскладок клавиатуры, применение регистровых клавиш и буфера обмена Windows были представлены выше

при описании стандартного приложения Блокнот. В данном разделе мы остановимся на особенностях текстового процессора Microsoft Word 2010, позволяющих автоматизировать ввод текста.

Средства отмены и возврата действий. Все операции ввода, редактирования и форматирования текста протоколируются текстовым процессором, и потому необходимое количество последних действий можно отменить. Последнее действие отменяют комбинацией клавиш **CTRL+Z**. Эта команда имеет кумулятивный эффект: серия команд отменяет серию последних действий. Другое аналогичное средство — кнопка **Отменить действие** на панели быстрого доступа. Длинные последовательности действий можно отменять также с помощью списка действий (кнопка, раскрывающая список, присоединена к кнопке **Отменить действие**).

После отмены ряда действий существует возможность вернуться к состоянию, предшествовавшему отмене, воспользовавшись клавишей **F4** на клавиатуре. Для этого также служит кнопка **Вернуть действие** на панели быстрого доступа.

Расширенный буфер обмена. При компиляции документа путем использования фрагментов текста, взятых из разных первоисточников, удобно пользоваться расширенным буфером обмена. Расширенный буфер обмена может хранить до 24 объектов (рис. 10.4.).

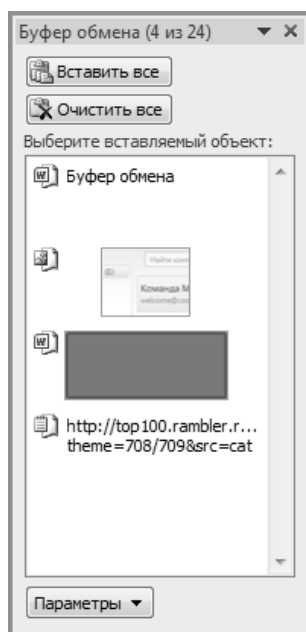


Рис. 10.4. Буфер обмена

Для того чтобы открыть Область задач в режиме Буфер обмена, можно воспользоваться кнопкой **Буфер обмена** в одноименной группе элементов управления на вкладке **Главная**. При необходимости можно настроить параметры автоматического отображения панели Буфер обмена.

Используя кнопку **Параметры** на панели **Буфер обмена**, можно задать особенности ее автоматического отображения. Содержание конкретного элемента буфера можно просмотреть в **Области задач**. В частности, тип содержимого элемента можно определить по значку, находящемуся рядом с ним, а также по внешнему виду самого элемента. При переполнении расширенного буфера самый старый элемент теряется, а очередной поступает в освобожденную ячейку.

Область задач позволяет вставить любой из имеющихся элементов, а также скомпоновать их в единый объект и вставить все сразу.

Автотекст — это режим автоматического ввода фрагментов текста. Он представлен двумя функциями: *автозавершением* и собственно *автотекстом*. Их принцип действия состоит в следующем.

Текстовый процессор хранит *словарь автотекста*, состоящий из слов и фраз, встречающихся в документах достаточно часто. При вводе первых четырех символов словарного элемента на экране появляется всплывающая подсказка с полным текстом слова или фразы. Если это то, что имел в виду пользователь, он завершает ввод всего фрагмента нажатием клавиши **Enter** — так работает функция *автозавершения*. Однако пользователь может самостоятельно выбрать необходимый элемент текста из списка с иерархической структурой — это функция *автотекста*. Список элементов автотекста можно найти по адресу **Вставка ► Текст ► Экспресс-блоки**.

Настройку словаря автотекста выполняют в диалоговом окне **Создание нового стандартного блока (Вставка ► Текст ► Экспресс-блоки ► Автотекст ► Сохранить выделенный фрагмент в коллекцию автотекста)**. Простейший способ наполнения словаря новым содержанием — выделить текст на экране и выполнить вышеуказанную команду.

Использование средства автозамены при вводе. Microsoft Word позволяет эффективно сократить объем вводимого текста за счет использования средства **Автозамена**. Оно позволяет заменить ввод длинных последовательностей символов произвольным (желательно коротким) сочетанием других символов. Например, если в тексте очень часто встречается словосочетание «диалоговое окно», его можно заменить коротким сочетанием «до». Соответственно вместо «диалоговых окон» использовать «дн», а вместо «диалогового окна» — «да». Точка перед символами стоит специально, чтобы отличать их от двухбуквенных предлогов или союзов, таких как «да».

Настройку средства Автозамена выполняют в диалоговом окне **Файл ► Параметры ► Правописание ► Параметры автозамены**.

Для этого надо установить флажок **Заменять при вводе**, ввести заменяемую комбинацию в поле **Заменить**, а замещающую комбинацию в поле **На**, после чего пополнить список автозамены щелчком на кнопке **Добавить**.

Как будет показано ниже, средство автоматической замены символов при вводе используется также для ввода специальных символов. Например, выполнив соответствующие настройки, можно вводить греческие буквы π и ρ обычным русским текстом: «пи» или «ро».

Ввод специальных и произвольных символов. При вводе текста часто существует необходимость ввода специальных символов, не имеющих соответствующей клавиши в раскладке клавиатуры, а также произвольных символов, раскладка для

которых неизвестна. Основным средством для ввода специальных и произвольных символов, а также для закрепления их за избранными клавишами является диалоговое окно **Символ** (**Вставка** ► **Символы** ► **Символ** ► **Другие символы**). Данное диалоговое окно имеет две вкладки: **Символы** и **Специальные знаки**.

На вкладке **Специальные знаки** присутствует список специальных символов, таких как «длинное» («полиграфическое») тире, «копирайт», «торговая марка» и др. Для вставки такого символа достаточно щелкнуть на кнопке **Вставить**. Вместе с тем, для большинства специальных символов существуют клавиатурные комбинации — они приведены в списке и их стоит запомнить. На первых порах, пока навык их ввода не закреплен, это окно используют для получения справки. В том же окне имеются кнопки **Автозамена** и **Сочетание клавиш**, позволяющие либо выполнять ввод специальных символов обычными символами и автоматически производить замену, либо закрепить специальный символ за избранной комбинацией клавиш.

На вкладке **Символы** представлены элементы управления для ввода произвольных символов любых символьных наборов (рис. 10.5). Центральное положение в окне занимает таблица символов текущего набора. Выбор шрифта выполняют в раскрывающемся списке **Шрифт**. Если шрифт относится к категории универсальных шрифтов **UNICODE**, то для него имеется и возможность выбора символьного набора в раскрывающемся списке **Набор**.

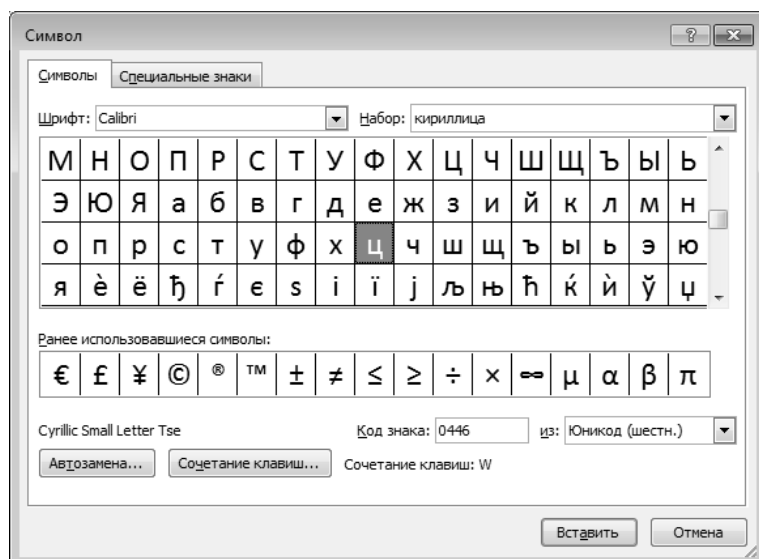


Рис. 10.5. Средство ввода специальных символов

Если символ надо вставить только один раз, достаточно щелкнуть на командной кнопке **Вставить**. Если предполагается многократное использование данного символа, за ним можно закрепить постоянную комбинацию клавиш (кнопка **Сочетание клавиш**) или создать элемент для списка **Автозамена** с помощью одноименной кнопки.

Специальные средства редактирования текста

Базовые приемы редактирования текста мы рассмотрели в разделе, посвященном стандартному приложению Блокнот. В данном разделе мы рассмотрим специальные средства редактирования, характерные для текстового процессора Microsoft Word, на примере последней версии Microsoft Word 2010.

Режимы вставки и замены символов. Текстовый процессор предоставляет возможность выбора между двумя режимами редактирования текста: *режимом вставки* и *режимом замены*. В режиме вставки вводимый текст «раздвигает» существующий текст, а в режиме замены новые символы замещают символы предшествующего текста, находившиеся в точке ввода. Режим вставки применяют при разработке основных содержательных блоков текстовых документов, а режим замены — при редактировании стандартных форм и стандартных элементов (колонтитулов, реквизитных элементов в письмах, служебных записках, бланках).

Для того чтобы отобразить в строке состояния индикатор режима работы программы, нужно щелкнуть по строке состояния правой кнопкой мыши и в появившемся контекстном меню установить флажок напротив параметра Замена. После этого режимы работы можно будет переключать, щелкая мышью по индикатору, который может принимать значения Замена и Вставка.

Настройка режима правки выполняется по адресу Файл ► Параметры ► Дополнительно ► Параметры правки.

Если установлен флажок Использовать клавишу INS для переключения режимов вставки и замены, режимы можно переключать нажатием на клавишу INS на клавиатуре. Возможность изменять режим путем щелчка на индикаторе в строке состояния сохраняется в любом случае.

По умолчанию активирован режим вставки, при необходимости постоянно работать в режиме замены следует установить флажок Использовать режим замены.

Использование Тезауруса. Тезаурус представляет собой словарь смысловых синонимов. При подготовке технической документации особую роль играют смысловые синонимы к используемым глаголам. Для выделенного слова тезаурус удобно вызывать через пункт Синонимы контекстного меню. Однако этот прием срабатывает далеко не для всех слов (преимущественно для глаголов в неопределенной форме). Общий прием вызова тезауруса состоит в использовании команды строки меню Рецензирование ► Правописание ► Тезаурус.

Окно Тезаурус имеет поле для ввода искомого слова и поле для вывода синонимов.

Кроме синонимов в некоторых случаях тезаурус позволяет находить *антонимы* слов и *связанные* (как правило, однокоренные) слова.

Средства автоматизации проверки правописания. Средства автоматизации проверки правописания включают средства проверки орфографии и грамматики. Текстовый процессор позволяет реализовать два режима проверки правописания — *автоматический* и *командный*.

Для работы в автоматическом режиме надо установить флажки Автоматически проверять орфографию и Автоматически проверять грамматику на вкладке Правописание диалогового окна Параметры Word (Файл ► Параметры ► Правописание). В автоматическом режиме слова, содержащие орфографические ошибки, подчер-

квиваются красным цветом, а выражения, содержащие грамматические ошибки, — зеленым. Для того чтобы узнать характер ошибки, надо щелкнуть правой кнопкой мыши на помеченном фрагменте. В зависимости от характера ошибки контекстное меню содержит пункт Орфография или Грамматика. С их помощью открывается диалоговое окно, в котором имеются элементы управления для получения более точной справки о том, какое правило нарушено, и предложены варианты исправления предполагаемой ошибки.

Встроенное автоматическое средство проверки правописания является, по существу, экспертной системой и допускает настройку. Так, например, если рекомендации экспертной системы неточны или неприемлемы, от них можно отказаться командой Пропустить (обычно такое бывает при проверке грамматики). Если же слово отмечено как орфографическая ошибка только потому, что оно отсутствует в словаре системы автоматической проверки (например, слово браузер), то его можно добавить в словарь.

Встроенный словарь системы проверки правописания не подлежит правке. Все дополнения и изменения вносятся в специальный подключаемый *пользовательский словарь*. Каждый пользователь может создать несколько специализированных пользовательских словарей, ориентированных на различные области знаний (автомобильное дело, машиностроение, вычислительная техника и т. п.). Подключение нужного словаря для работы с конкретным документом выполняется выбором словарного файла в диалоговом окне Вспомогательные словари. Чтобы открыть его, надо щелкнуть на кнопке Словари на вкладке Файл ► Параметры ► Правописание. Постепенно наполняясь конкретным содержанием, вспомогательные словари пользователя становятся мощным средством повышения производительности его труда.

В командном режиме проверка правописания выполняется независимо от установки элементов управления на вкладке Файл ► Параметры ► Правописание. Запуск средства проверки выполняют командой Рецензирование ► Правописание ► Правописание. Проверка начинается от местоположения курсора и продолжается до появления первой ошибки. После исправления ошибки проверка продолжается дальше, а по достижении конца документа проверка может быть продолжена с его начала. Естественное завершение проверки происходит, когда документ просмотрен целиком.

В тех случаях, когда пользователь отказывается от предлагаемых исправлений и дает команду Пропустить, в документе накапливается *список пропускаемых слов*, то есть слов и выражений, не подлежащих проверке. Для того чтобы очистить этот список и начать проверку заново, используют командную кнопку Файл ► Параметры ► Правописание ► Повторная проверка.

Средства рецензирования текста

Под рецензированием можно понимать два процесса: *редактирование текста с регистрацией изменений* и *комментирование* текста. В отличие от обычного редактирования при рецензировании текст документа изменяется не окончательно —

новый вариант и старый «сосуществуют» в рамках одного документа на правах различных *версий*.

Основным средством рецензирования является вкладка Рецензирование. На ней представлены несколько групп элементов управления, предназначенных для:

- ☐ создания, просмотра и удаления примечаний;
- ☐ регистрации, просмотра, принятия и отмены изменений;
- ☐ выбора цвета выделения примечаний;
- ☐ сохранения версий документа;
- ☐ сравнения документов;
- ☐ проверки правописания.

Для создания примечания служит кнопка Создать примечание. При ее использовании местоположение курсора выделяется заданным цветом, а на правом поле страницы открывается область ввода текста примечания. Созданное примечание отображается только при просмотре документа, но не при его печати.

Для регистрации изменений в тексте служит кнопка Исправления. Все редактирование текста в режиме регистрации исправлений считается неавторским и выделяется особым методом (метод выделения можно задать в диалоговом окне Параметры исправлений (Рецензирование ► Запись исправлений ► Исправления ► Параметры исправлений)).

Элементы управления группы элементов Изменения позволяют выполнять переходы между исправлениями, принимать их или отвергать.

Если документ проходит многоступенчатое редактирование, часто возникает необходимость хранить его промежуточные версии. Текстовый процессор Microsoft Word XP позволяет хранить несколько версий документа в одном файле. Это удобное средство отличается тем, что при сохранении нескольких версий (в отличие от нескольких копий) эффективно используется рабочее место на диске. Дело в том, что при сохранении очередной версии не происходит повторного сохранения всего документа — сохраняются только отличия текущей версии от предшествующей. Для сохранения текущей версии и для загрузки одной из промежуточных версий применяют команду Файл ► Сведения ► Управление версиями.

Форматирование текста

Форматирование текста осуществляется средствами контекстного меню выделенного фрагмента или при помощи средств вкладок ленты. В частности, речь идет о вкладках Главная и Разметка страницы, где сосредоточены наиболее часто используемые инструменты для форматирования текста.

- ☐ выбор и изменение гарнитуры шрифта;
- ☐ управление размером шрифта;
- ☐ управление начертанием и цветом шрифта;
- ☐ управление методом выравнивания;

- ☐ создание маркированных и нумерованных списков (в том числе многоуровневых);
- ☐ управление параметрами абзаца.

Настройка шрифта. При выборе гарнитуры шрифта необходимо учитывать следующие обстоятельства:

- ☐ Выбор гарнитуры шрифта действует на выделенный текстовый фрагмент. Если ни один фрагмент не выделен, он действует на весь вводимый текст до очередной смены гарнитуры.
- ☐ Microsoft Word ориентирован на работу с многоязычными шрифтовыми наборами (UNICODE). При использовании других шрифтовых наборов возможны проблемы, которые возникают при переключении раскладки клавиатуры с основной (английской) на дополнительную (русскую). В таком случае возможен неконтролируемый автоматический возврат к использованию одного из стандартных шрифтов UNICODE, зарегистрированных в операционной системе.

Настройку шрифта выполняют в группе параметров Шрифт (Главная ► Шрифт). Более детально параметры шрифта можно выполнить в диалоговом окне Шрифт, которое можно вызвать нажатием на кнопку в правом нижнем углу группы параметров Шрифт. Данное диалоговое окно имеет две вкладки: Шрифт и Дополнительно.

На вкладке Шрифт выбирают:

- ☐ гарнитуру шрифта;
- ☐ его размер (измеряется в полиграфических пунктах);
- ☐ вариант начертания;
- ☐ цвет символов;
- ☐ наличие подчеркивания;
- ☐ характер видоизменения.

При выборе гарнитуры шрифта следует иметь в виду, что существует две категории шрифтов: с засечками и без засечек (рубленые). Характерными представителями первой категории являются шрифты семейства *Times*, а второй категории — шрифты семейства *Arial*. Шрифты, имеющие засечки, легче читаются в больших текстовых блоках — их рекомендуется применять для оформления основного текста.

Шрифты, не имеющие засечек, рекомендуется использовать для заголовков в технических текстах, а также для оформления дополнительных материалов (врезок, примечаний и пр.).

Кроме того, считается, что шрифты с засечками лучше воспринимаются в документах, напечатанных на бумаге. Для электронных документов, которые предполагается читать с экрана, многие предпочитают применять рубленые шрифты.

Большинство гарнитур шрифтов являются *пропорциональными*. Это означает, что и ширина отдельных символов, и расстояние между соседними символами не являются постоянными величинами и динамически меняются так, чтобы сопряжение символов было наиболее благоприятным для чтения.

Особую группу представляют так называемые *моноширинные* шрифты. В них каждый символ вместе с окаймляющими его интервалами имеет строго определенную ширину. Такие шрифты применяют в тех случаях, когда надо имитировать шрифт пишущей машинки, а также при вводе текстов, представляющих листинги программ. Характерными представителями таких шрифтов являются шрифты семейства *Courier*.

При выборе размера шрифта руководствуются назначением документа, а также вертикальным размером печатного листа. Для документов, имеющих формат типовой книжной страницы, обычно применяют шрифт размером 10 пунктов. Для документов, готовящихся для печати на стандартных листах формата A4 (210 × 297 мм), выбирают размер 12 пунктов. При подготовке документов, предназначенных для передачи средствами факсимильной связи, применяют увеличенный размер — 14 пунктов и больше (факсимильные документы часто воспроизводятся с искажениями, и увеличенный размер шрифта улучшает удобство их чтения).

При подготовке электронных документов, распространяемых в формате Microsoft Word, размер шрифта выбирают исходя из разрешения экрана. В настоящее время наиболее распространены компьютеры, видеоподсистема которых настроена на экранное разрешение от 1024 × 768 точек. Для этих параметров целесообразно готовить электронные документы с размером шрифта 11 пунктов. На этот размер по умолчанию настроен Microsoft Word 2010.

Использование прочих средств управления шрифтом (выбор начертания, подчеркивания и других видоизменений) определяется стилевым решением документа, которое задает заказчик или работодатель. Приступая к первому заданию, следует выяснить, какие стилевые решения уже существуют в данной организации, каковы ограничения на использование средств оформления и форматирования. По возможности, надо получить от заказчика готовые шаблоны документов или хотя бы печатные образцы.

Из прочих, не рассмотренных здесь средств управления шрифтами надо отметить управление интервалом между символами и возможность использования эффектов анимации. Интервал задается путем выбора одного из трех значений (Обычный, Разреженный, Уплотненный) на вкладке Дополнительно окна Шрифт.

Эффекты анимации (Главная ► Шрифт ► Параметры анимации) используют очень редко и только при подготовке электронных документов, распространяемых в формате текстового процессора.

Настройка метода выравнивания. Все последние версии текстового процессора Microsoft Word поддерживают четыре типа выравнивания:

- ☐ по левому краю;
- ☐ по центру;
- ☐ по правому краю;
- ☐ по ширине.

Выбор метода выполняют соответствующими кнопками группы инструментов Абзац вкладки Главная или посредством окна Абзац, которое можно вызвать из контекстного меню выделенного участка текста или воспользовавшись кнопкой Абзац одноименной группы инструментов.

Избранный метод действует на текущий и последующие вводимые абзацы. Выбор метода выравнивания определяется назначением документа.

Для документов, передаваемых на последующую обработку, все методы выравнивания, кроме тривиального выравнивания по левому краю, являются излишними. Для печатных документов, выполненных на русском или немецком языках, рекомендуется в основном тексте использовать выравнивание по ширине с одновременным включением функции переноса, а для документов на английском языке основной метод выравнивания — по левому полю.

Настройка параметров абзаца. Кроме режима выравнивания настраиваются следующие параметры абзаца:

- ☐ величина отступа слева (от левого поля);
- ☐ величина отступа справа (от правого поля);
- ☐ величина отступа первой строки абзаца («красная строка»);
- ☐ величина интервала (отбивки между абзацами) перед абзацем и после него.

Для печатных документов величину отступа для основного текста, как правило, не задают (необходимое положение текста определяется шириной полей), но ее задают для дополнительных материалов и заголовков, если они не выравниваются по центру. В то же время для веб-страниц величина отступа для абзацев имеет большое значение.

Роль отбивок между абзацами, как и роль отступа первой строки абзаца, состоит в том, чтобы визуально выделить абзацы. При этом следует помнить, что эти средства несовместимы. То есть, применяя отступ первой строки абзаца, не следует применять отбивки между абзацами, и наоборот. Комбинация этих стилей допускается только для маркированных и нумерованных списков (основной текст оформляется с отступом первой строки, а списки — без него, но с отбивкой между абзацами).

Обычная практика назначения формата состоит в том, что для документов простой структуры (художественных) используют отступ первой строки (это особенно важно для текстов на русском и немецком языках), а для документов сложной структуры (технических) и документов на английском языке используют отбивки между абзацами. Промежуточное положение занимают документы, относящиеся к естественно-научным и гуманитарным дисциплинам, — при их подготовке, кроме точки зрения автора, руководствуются сложившейся практикой и устоявшимися традициями.

В веб-документах применяют только отбивки между абзацами. Отступ первой строки в них обычно не используют.

Средства создания маркированных и нумерованных списков. Специальное оформление маркированных и нумерованных списков редко применяют в художественных документах и персональной переписке, но в служебных документах и особенно в веб-документах оно используется очень широко. В веб-документах оформление маркированных списков особо усиливают за счет применения специальных графических маркеров, стиль которых должен тематически сочетаться с содержанием и оформлением документов.

Для создания нумерованных и маркированных списков нужно сначала выполнить *настройку*, затем *вход* в список и, наконец, *выход* из него. Настройку

выполняют в группе параметров **Абзац** вкладки **Главная** с помощью кнопок **Маркеры**, **Нумерация** и **Многоуровневый список**. При необходимости можно вызвать меню каждой из этих кнопок для выбора варианта списка и настройки его дополнительных параметров

Вход в список может осуществляться автоматически или по команде. Чтобы автоматически создать маркированный список, достаточно начать запись строки с ввода символа «*». По завершении строки и нажатии клавиши **ENTER** символ «*» автоматически преобразуется в маркер, а на следующей строке маркер будет установлен автоматически. Для автоматического создания нумерованного списка достаточно начать строку с цифры, после которой стоят точка и пробел, например «1. », «2. » и т. д. Этот метод позволяет начать нумерацию с любого пункта (не обязательно с единицы).

Как маркированный, так и нумерованный список легко превратить в многоуровневый. Для перехода на новые (или возврата на предшествующие) уровни служат кнопки **Увеличить отступ** и **Уменьшить отступ** в группе элементов **Абзац**.

Для списков с очень глубоким вложением уровней (более трех) можно настроить стиль оформления каждого из уровней. Для этого служит меню уже упомянутой командной кнопки **Многоуровневый список**.

Характерной особенностью процессора Microsoft Word 2010, связанной с его ориентацией на создание веб-документов, является возможность использования графических маркеров.

Для выбора такого маркера следует вызвать окно **Главная ▸ Абзац ▸ Маркеры ▸ Определить новый маркер** и в появившемся окне воспользоваться кнопкой **Рисунок** для выбора подходящего изображения.

Для завершения маркированного или нумерованного списка и выхода из режима его создания достаточно по завершении ввода последней строки дважды нажать клавишу **ENTER**.

10.3. Приемы и средства автоматизации разработки документов

С рядом приемов автоматизации ввода и редактирования текста мы познакомились выше. К ним относятся средства **Автотекст**, **Автозамена**, средства проверки правописания, средства расстановки переносов, средства поиска и замены фрагментов текста.

В этом разделе мы познакомимся с наиболее общими средствами автоматизации разработки и оформления документов, к числу которых относятся *стили оформления абзацев, шаблоны документов и темы оформления*.

Работа со стилями

Абзац — элементарный элемент оформления любого документа. Каждый заголовок документа тоже рассматривается как отдельный абзац. Выше мы видели, что в группе параметров **Главная ▸ Абзац** имеется немало различных элементов управления, и выполнять их настройку для каждого абзаца отдельно — неэффективная и утомительная задача. Она автоматизируется путем использования понятия *стиль*.

Стиль оформления — это именованная совокупность настроек параметров шрифта, абзаца, языка и некоторых элементов оформления абзацев (линий и рамок). Благодаря использованию стилей обеспечивается простота форматирования абзацев и заголовков текста, а также единство их оформления в рамках всего документа.

Особенностью текстовых процессоров Microsoft Word является то, что они поддерживают четыре типа стилей: *стили абзаца*, *стили знаков* (символов), *стили списков* и *стили таблиц*. С помощью стилей абзаца выполняют форматирование абзацев, а с помощью знаковых стилей можно изменять оформление выделенных фрагментов текста внутри абзаца. Стиль списка предполагает наличие в начале абзаца номера или маркера. Стиль таблицы обеспечивает согласование границ, заливки, выравнивания и шрифтов в таблицах.

Наличие разных типов стилей позволяет реализовать довольно сложные приемы форматирования. Например, внутри абзаца, оформленного одним шрифтом, могут содержаться фрагменты текста, оформленные другим шрифтом. В данной книге, например, специальный шрифт использован для записи названий элементов управления.

Работа со стилями состоит в создании, настройке и использовании стилей. Некоторое количество стандартных стилей присутствует в настройке программы по умолчанию, сразу после ее установки. Их используют путем выбора нужного стиля в группе элементов **Стили** вкладки **Главная**.

Все работы по созданию новых стилей и изменению существующих выполняют с помощью **Области задач** в режиме **Стили**. Для ее вызова нужно воспользоваться кнопкой **Стили**, которая расположена в правом нижнем углу группы элементов **Стили**.

Настройка стиля. Настройку стиля (рис. 10.6) выполняют в диалоговом окне **Создание стиля** **Главная** ► **Стили** ► **Стили** ► **Создать стиль**. Настраиваемый стиль выбирают в списке **Стили**. Для изменения стиля служит команда **Изменить контекстного меню** выбранного стиля, открывающая диалоговое окно **Изменение стиля**.

При проведении настройки стиля важно правильно выбрать исходный стиль. Он должен быть как можно ближе к желаемому, чтобы минимизировать количество необходимых настроек.

Создание стиля. Для создания нового стиля надо щелкнуть на кнопке **Создать стиль** в окне **Стили**. Откроется диалоговое окно **Создание стиля**.

В данном окне следует:

- ☐ ввести название нового стиля в поле **Имя**;
- ☐ выбрать тип стиля (стиль абзаца, знака, списка или таблицы);
- ☐ выбрать стиль, на котором основан новый стиль;
- ☐ указать стиль следующего абзаца;
- ☐ настроить основные элементы стиля, используя средства данного диалогового окна;
- ☐ настроить дополнительные элементы стиля с помощью кнопки **Формат**.

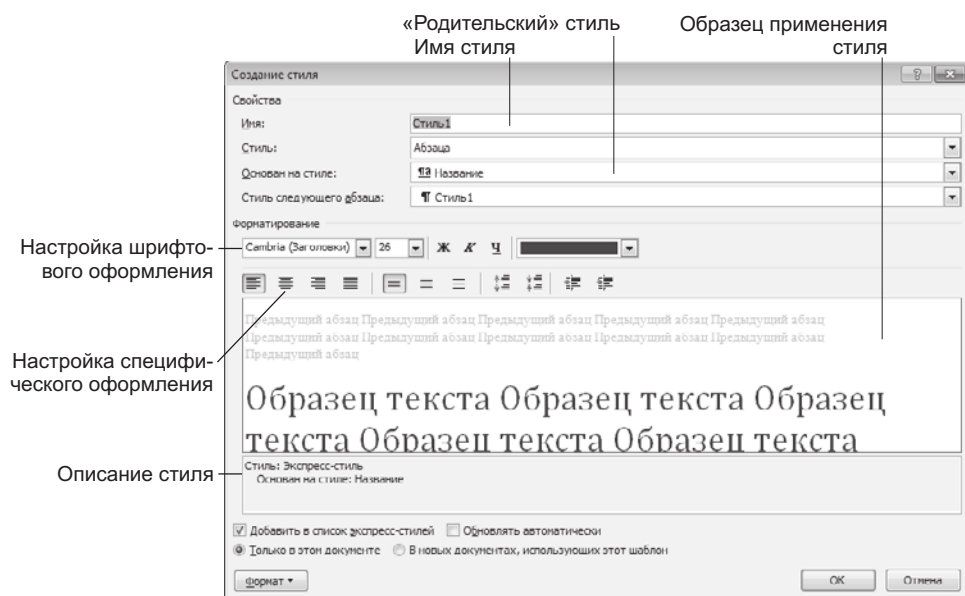


Рис. 10.6. Настройка стиля

Важной чертой программы является принцип *наследования стилей*. Он состоит в том, что любой стиль может быть основан на каком-то из существующих стилей. Это позволяет, во-первых, сократить настройку стиля до минимума, сосредоточившись только на отличиях от базового, а во-вторых, обеспечить принцип единства оформления всего документа в целом. Так, например, при изменении базового стиля автоматически произойдут и изменения наследуемых элементов в стилях, созданных на его основе.

Стиль следующего абзаца указывают для обеспечения автоматического применения стиля к следующему абзацу, после того как предыдущий абзац закрывается клавишей **ENTER**.

Разработка новых стилей и их настройка являются достаточно сложными технологическими операциями. Они требуют тщательного планирования, внимательности и аккуратности, особенно в связи с тем, что согласно принципу наследования свойств стилей желаемые изменения в одном стиле могут приводить к нежелательным изменениям во многих других стилях.

В связи с трудоемкостью изучения и освоения приемов практической работы со стилями начинающие пользователи часто ими пренебрегают. Действительно, при разработке небольших документов (одна-две страницы) можно обойтись без настройки и использования стилей, выполнив все необходимое форматирование вручную средствами вкладки **Главная**. Однако при разработке объемных документов вручную очень трудно обеспечить единство оформления, особенно если разные разделы документа разрабатывались разными авторами.

Прийти к использованию стилей надо как можно раньше. Правильное и рациональное использование этого средства является залогом высокой эффективности работы с процессором Microsoft Word и высокого качества разрабатываемых документов. На изучение средств управления стилями может потребоваться несколько часов, но полученные при этом навыки останутся на всю жизнь и пригодятся многократно.

Шаблоны

Совокупность удачных стилевых настроек сохраняется вместе с готовым документом, но желательно иметь средство, позволяющее сохранить их и вне документа. Тогда их можно использовать для подготовки новых документов. Такое средство есть — это шаблоны, причем некоторое количество универсальных шаблонов поставляется вместе с текстовым процессором и устанавливается на компьютере вместе с ним.

По своей сути шаблоны — это тоже документы, а точнее говоря, заготовки будущих документов. От обычных документов шаблоны отличаются тем, что в них приняты специальные меры, исключающие возможность их повреждения. Открывая шаблон, мы начинаем новый документ и вносим изменения в содержание шаблона. При сохранении же мы записываем новый документ, а шаблон, использованный в качестве его основы, остается в неизменном виде и пригоден для дальнейшего использования.

Использование шаблона для создания документа. По команде **Файл ▶ Создать** открывается список имеющихся в системе шаблонов. Здесь можно выбрать шаблон, на базе которого документ будет разрабатываться. В этом случае документ сразу получает несколько готовых стилей оформления, которые содержатся в шаблоне. Если представленных в окне шаблонов недостаточно, можно просмотреть дополнительные шаблоны, например, открываемые по щелчку на значке **Образцы шаблонов**. Можно воспользоваться поиском шаблонов на office.com.

Изменение шаблона готового документа. Эта достаточно редкая операция выполняется с помощью диалогового окна **Шаблоны и надстройки (Разработчик ▶ Надстройки ▶ Надстройки)**. Для смены текущего шаблона следует использовать кнопку **Присоединить** и в открывшемся диалоговом окне **Присоединение шаблона** выбрать нужный шаблон.

Создание нового шаблона на базе шаблона. Выполнив команду **Файл ▶ Создать**, следует выбрать шаблон для создания документа. После настройки стилей и редактирования содержания выполняется сохранение шаблона командой **Сохранить** с выбором пункта **Шаблон Word** или **Шаблон Word с поддержкой макросов** в поле **Тип файла**.

Создание нового шаблона на базе документа. Если готовый документ может быть использован в качестве заготовки для создания других документов, его целесообразно сохранить как шаблон. Командой **Файл ▶ Открыть** открывают готовый документ, в нем правят содержание и настраивают стили, а потом сохраняют документ как шаблон командой **Сохранить как** с включением пункта **Шаблон Word** или **Шаблон Word с поддержкой макросов** в поле **Тип файла**.

Темы

Microsoft Word имеет специальное средство автоматического оформления, предназначенное в первую очередь для электронных документов (для веб-документов и документов, распространяемых в формате процессора). Это средство называется *темы*. Тема представляет собой совокупность следующих элементов оформления:

- ☐ фоновый узор;
- ☐ стили оформления основного текста и заголовков;
- ☐ стиль оформления маркированных списков;
- ☐ стиль графических элементов оформления (линий).

При выборе темы оформления становятся доступными стили оформления, указанные для данной темы.

Доступ к выбору тем выполняется командой Разметка страницы ► Темы.

Практическое занятие

Упражнение 10.1. Первичные настройки текстового процессора Microsoft Word 2010

1. Запустите текстовый процессор командой Пуск ► Все программы ► Microsoft Office ► Microsoft Word 2010.
2. Откройте заранее подготовленный файл (произвольный).
3. Откройте меню настройки ленты (Файл ► Параметры ► Настройка ленты) и включите отображение вкладки Разработчик.
4. В качестве режима отображения документа выберите Режим разметки. Для этого используйте соответствующую кнопку в правом нижнем углу окна документа или команду Вид ► Режимы просмотра документа ► Разметка страницы.
5. Если шрифт на экране выглядит слишком мелким, настройте масштаб отображения командами группы параметров Масштаб вкладки Вид. Можно также использовать полосу прокрутки Масштаб в правом нижнем углу окна документа. Для эффективного использования площади окна документа при достаточном разрешении экрана можно использовать пункты По ширине страницы или По ширине текста окна Масштаб (Вид ► Масштаб ► Масштаб).
6. В качестве единицы измерения для настройки параметров документа выберите миллиметры (Файл ► Параметры ► Дополнительно ► Экран ► Единица измерения).
7. Настройте список быстрого открытия документов. После запуска программы в меню Файл ► Последние можно найти список из нескольких документов, открывавшихся в текстовом процессоре в последнее время. Это удобно для быстрого открытия нужного документа. Количество документов, отображаемых в этом списке, задайте счетчиком Файл ► Параметры ► Дополнительно ► Экран ► Число документов в списке последних файлов.
8. Отключите замену выделенного фрагмента при правке текста, сбросив флажок Файл ► Параметры ► Дополнительно ► Параметры правки ► Заменять выделенный

фрагмент. Это несколько снижает производительность труда при редактировании текста, но страхует начинающих от нежелательных ошибок. С набором опыта практической работы этот флажок можно установить вновь.

9. Включите контекстно-чувствительное переключение раскладки клавиатуры (Файл ► Параметры ► Дополнительно ► Параметры правки ► Автоматически переключать раскладку клавиатуры в соответствии с языком окружающего текста). Эта функция удобна при редактировании текста. При помещении курсора в английский текст автоматически включается англоязычная раскладка, а при помещении его в текст на русском языке — русскоязычная.
10. Настройте функцию *автосохранения* с помощью счетчика Файл ► Параметры ► Сохранение ► Автосохранение каждые ... минут. Имейте в виду следующие обстоятельства:
 - при автосохранении данные записываются в специальный файл, который в аварийных ситуациях может быть использован для восстановления не-сохраненных данных, но только однократно(!);
 - функция автосохранения не отменяет необходимости периодически во время работы и после ее завершения сохранять файл прямыми командами Сохранить или Сохранить как.
11. Временно отключите средства проверки правописания. На вкладке Файл ► Параметры ► Правописание сбросьте флажки Автоматически проверять орфографию и Автоматически проверять грамматику. На ранних этапах работы с документом надо сосредоточиться на его содержании, а средства проверки правописания могут отвлекать от этого. Завершая работу над документом, необходимо вновь подключить и использовать эти средства.
12. Временно отключите функцию *автозамены при вводе* сбросом флажка Файл ► Параметры ► Правописание ► Параметры автозамены ► Заменять при вводе.
13. Включите автоматическую замену «прямых» кавычек парными: Файл ► Параметры ► Правописание ► Параметры автозамены ► Автоформат при вводе ► Заменять при вводе «прямые» кавычки парными. В русскоязычных текстах прямые кавычки не применяются. Для подготовки англоязычных текстов и листингов программ отключите эту функцию.
14. Временно отключите ряд средств автоматического форматирования, в частности автоматическую маркировку и нумерацию списков. На вкладке Файл ► Параметры ► Правописание ► Параметры автозамены ► Автоформат при вводе сбросьте флажки Применять при вводе стили маркированных списков и Применять при вводе стили нумерованных списков. После приобретения первичных навыков работы с текстами вновь подключите эти средства.
15. Отключите автоматическую расстановку переносов. В абсолютном большинстве случаев на ранних этапах работы с документами она не нужна. Для веб-документов, для документов, распространяемых в формате текстового процессора, и для документов, передаваемых на последующую обработку, расстановка переносов не только бесполезна, но и вредна. Для документов, которые окончательно форматируются и распечатываются в одной рабочей

группе, расстановка переносов может быть полезной, но и в этом случае ее применяют только на заключительных этапах форматирования и при этом очень тщательно проверяют соответствие переносов, расставленных автоматически, нормам и правилам русского языка.

Расстановку переносов отключают сбросом флажка Разметка страницы ► Параметры страницы ► Расстановка переносов ► Параметры расстановки переносов ► Автоматическая расстановка переносов.

16. Включите запрос на подтверждение изменения шаблона «Обычный»: Файл ► Параметры ► Дополнительно ► Сохранение ► Запрос на сохранение шаблона Normal.dot. Шаблон «Обычный» является первоосновой для всех остальных шаблонов (они создаются на его базе и наследуют его свойства). При обычной работе с программой необходимость его изменения не возникает (если надо что-то изменить в этом шаблоне, достаточно создать его копию под другим именем и работать с ней). Включением данного флажка предупреждаются случайные внесения изменений в шаблон со стороны пользователя, а также попытки макровирусов сохранить свой код в данном шаблоне (для дальнейшего размножения в документах, создаваемых на его основе).

Итог. Мы научились выполнять первичные настройки текстового процессора и узнали, что доступ к ним осуществляется через различные вкладки окна, вызываемого командой Файл ► Параметры, а также некоторыми командами с вкладок ленты.

Упражнение 10.2. Первичные настройки параметров печатного документа

Форматирование документов, предназначенных для печати на принтере, выполняется в «привязке» к параметрам печатной страницы. Поэтому создание документов этой категории необходимо начинать с настройки параметров страницы. К этим параметрам относятся, прежде всего, размер листа бумаги и величина полей.

Характерная ошибка начинающих заключается в том, что они начинают подготовку документов с ввода текста. Интуитивно понятно, что текст — это важнейший компонент документа, но для ввода текста служат программы иного класса — текстовые редакторы. Имея дело с текстовым процессором, начинать надо не с ввода текста документа, а с настройки параметров печатной страницы, поскольку от нее зависят все используемые приемы форматирования. Тем, кому утомительно начинать создание каждого документа с настройки параметров страницы, можно порекомендовать чаще пользоваться заранее заготовленными шаблонами.

1. Запустите текстовый процессор командой Пуск ► Все Программы ► Microsoft Office ► Microsoft Word. Автоматически будет создан новый докумен.
2. Откройте диалоговое окно Параметры страницы (Файл ► Печать ► Параметры страницы).
3. На вкладке Размер бумаги выберите в раскрывающемся списке Размер бумаги пункт A4 (этот формат принят в России в качестве стандартного). При исполь-

зовании нестандартного формата выбирают пункт Другой и с помощью кнопок счетчиков Ширина и Высота задают его параметры.

4. На вкладке Поля задайте ориентацию бумаги (Книжная или Альбомная). При «альбомной» ориентации бумага располагается длинной стороной по горизонтали.
5. На этой же вкладке задайте размеры полей:
Верхнее — 15 мм Нижнее — 20 мм
Левое — 25 мм Правое — 15 мм
6. На вкладке Источник бумаги задайте для нижнего поля интервал от края до колонтитула 12 мм (в нижнем колонтитуле будет размещаться номер печатной страницы).
7. Если предполагается двусторонняя печать (четные страницы печатаются на оборотной стороне нечетных страниц), выберите на вкладке Поля пункт Зеркальные поля в списке Несколько страниц. Восстановите обычную настройку.
8. Проверьте, как действует настройка печати двух страниц на одном листе. Выберите в списке Несколько страниц пункт 2 страницы на листе. На панели Образец рассмотрите результат настройки. Установите «альбомную» ориентацию страниц. Оцените результат настройки. Восстановите «книжную» ориентацию и печать одной страницы на листе.
9. Создайте нижний колонтитул для размещения номера печатной страницы. Дайте команду Вставка ► Колонтитулы ► Номер страницы ► Внизу страницы и в появившемся меню выберите подходящее расположение и стиль оформления номера страницы. Убедитесь в том, что в документе появились номера страниц.

Чтобы удалить колонтитулы по всему документу, надо воспользоваться соответствующими командами, которые имеются в контекстных меню кнопок Верхний колонтитул, Нижний колонтитул, Номера страниц.

10. Сохраните документ командой Сохранить как, дав ему имя Эксперимент и используя для сохранения библиотеку \Документы.

Итог. Мы научились создавать и настраивать печатные документы. В порядке эксперимента мы создали «пустой» документ, имеющий настроенные параметры страницы, стили, соответствующие шаблону «Обычный», и нижний колонтитул для размещения номеров печатных страниц. Мы готовы к наполнению данного документа текстовым содержанием с последующим редактированием и форматированием.

Упражнение 10.3. Ввод специальных символов

В этом упражнении мы рассмотрим пять приемов ввода символов греческого алфавита. Особо отметим, что это еще далеко не все возможные приемы для текстового процессора Microsoft Word. Упражнение будем выполнять вводом фразы: «Длина окружности равна $2\pi R$ ». Для подготовки к упражнению запустите текстовый процессор и воспользуйтесь документом, созданным по умолчанию.

1. *Замена шрифта.* Введите текст: «Длина окружности равна $2\pi R$ ». Выделите букву «р». На вкладке Главная раскройте список шрифтов и выберите символьный набор Symbol. Символ «р» заменится символом « π ».

Кроме того, доступ к списку шрифтов можно получить из контекстного меню выделенного фрагмента. Это наиболее стандартный прием. Им можно пользоваться во всех программах, имеющих средства для изменения шрифта, но для его применения нужно заранее знать, какой символ латинского шрифта соответствует нужному символу греческого шрифта, а это не всегда возможно.

2. *Классический подход.* Введите текст: «Длина окружности равна $2\pi R$ ». Выделите символ «х». Откройте программу Таблица символов (Пуск ► Все программы ► Стандартные ► Служебные ► Таблица символов). В окне этой программы выберите шрифт Symbol. В поле таблицы разыщите символ π , выделите его, щелкните на кнопке Выбрать и на кнопке Копировать. Вернитесь в окно Microsoft Word и комбинацией клавиш CTRL+V вставьте из буфера обмена скопированный символ на место выделенного.

Этот прием действует в большинстве программ. Его применяют, если заранее не известно, какому символу латинского шрифта соответствует необходимый символ.

3. *Использование стиля.* Если документ содержит много символов греческого алфавита, имеет смысл создать для них специальный *знаковый стиль*. На базе существующего знакового стиля, например стиля Основной шрифт абзаца, создайте новый знаковый стиль, например Греческий. Для этого откройте окно Стили (Главная ► Стили ► Стили) и нажмите кнопку Создать стиль. В диалоговом окне Создание стиля в поле Имя введите имя нового стиля, в раскрывающемся списке Стиль выберите пункт Знака и в списке Основан на стиле выберите базовый стиль. Если предполагается и в дальнейшем создание аналогичных документов, созданный стиль можно сохранить в шаблоне, установив флажок Добавить в список экспресс-стилей и выбрав переключатель В новых документах, использующих этот шаблон. После этого выберите символьный набор Symbol в раскрывающемся списке на панели Форматирование. В дальнейшем при необходимости ввода греческих букв достаточно на панели форматирования выбрать стиль Греческий.

Этот прием специфичен для программы Microsoft Word. Далеко не все текстовые редакторы и процессоры позволяют создавать знаковые стили — большинство используют только стили абзаца, применение которых изменяет шрифт во всем абзаце целиком.

4. *Применение «горячих клавиш».* Это самый эффективный прием. Нет более быстрого способа ввода нестандартных символов, чем ввод с помощью заранее назначенных клавиатурных комбинаций. Так, например, мы можем закрепить символ π за комбинацией клавиш CTRL+ALT+P и использовать ее всюду, где в этом возникает необходимость.

Используйте команду Вставка ► Символы ► Символ ► Другие символы — откроется диалоговое окно Символ. В списке Шрифт выберите шрифт Symbol. В таблице символов разыщите и выберите символ π . Щелкните на кнопке Со-

четание клавиш — откроется диалоговое окно *Настройка клавиатуры*. Убедитесь в том, что текстовый курсор находится в поле *Новое сочетание клавиш* (в таких случаях говорят, что *фокус ввода* принадлежит элементу управления *Новое сочетание клавиш*). Если это не так, переместите фокус ввода в нужное поле последовательными нажатиями клавиши **ТАВ**.

Когда фокус ввода находится в нужном поле, нажмите желаемую комбинацию клавиш, например **CTRL+ALT+P**. Обратите внимание на запись, появившуюся в поле, и щелкните на кнопке *Назначить*. Закройте открытые диалоговые окна и проверьте работу данной комбинации.

Обратите внимание на то, что для одного и того же символа можно назначать несколько комбинаций клавиш. Если нужно изменить назначение, следует в диалоговом окне *Настройка клавиатуры* выделить назначенную комбинацию и щелкнуть на кнопке *Удалить*. Если нужно, чтобы назначенная комбинация действовала во всех вновь создаваемых документах, ее можно сохранить в текущем шаблоне, выбрав его в раскрывающемся списке *Сохранить изменения*.

5. *Использование средства автозамены*. У метода «горячих клавиш» есть существенный недостаток: надо запоминать, какому символу какая комбинация соответствует. Если предполагается ввод множества нестандартных символов, удобно использовать средство автоматической замены символов при вводе.

Используйте команду *Вставка ► Символы ► Символ ► Другие символы* — откроется диалоговое окно *Символ*. В списке *Шрифт* выберите шрифт **Symbol**. В таблице символов разыщите и выберите символ **π**. Щелкните на кнопке *Автозамена* — откроется диалоговое окно *Автозамена*. В поле *Заменить* введите заменяемую комбинацию «.пи.». (Зачем символы «пи» оконтурены точками с двух сторон, выясните самостоятельно, экспериментируя с вводом выражения **2πR**.) Аналогичным образом можно организовать ввод и других символов: «.фи.», «.тау.», «.кси.» и т. д. Как видите, ничего не надо специально запоминать.

Итог. В текстовом процессоре *Microsoft Word*, как и во многих других приложениях *Windows*, одну и ту же операцию можно выполнить множеством разных способов. У каждого способа есть достоинства и недостатки. Пользователи опытным путем подбирают наиболее удобные для себя приемы. Выбор приема зависит от объема и характера выполняемой работы, а также от периодичности ее исполнения.

Глава одиннадцатая

Создание комплексных текстовых документов

В предыдущей главе мы рассмотрели приемы создания *простых* текстовых документов средствами текстового процессора Microsoft Word. К условной категории *простых* эти документы были отнесены только потому, что не содержали объектов, встроенных в текст. Соответственно, нами не были рассмотрены вопросы взаимодействия текста и встроенных объектов.

В этой главе мы рассмотрим приемы создания *комплексных* текстовых документов, содержащих специальные элементы оформления и встроенные объекты нетекстовой природы (формулы, таблицы, диаграммы, художественные заголовки, растровые и векторные иллюстрации, а также объекты мультимедиа).

11.1. Приемы управления объектами Microsoft Word

Особенности объектов Word

Текстовый процессор Word 2010 обладает развитой функциональностью по работе с объектами нетекстовой природы. Среди встроенных объектов могут быть стандартные объекты, созданные другими программами (рисунки, анимационные и звуковые клипы и многое другое), а также объекты, созданные средствами самого текстового процессора. В частности, программа позволяет создавать и встраивать геометрические фигуры, художественные заголовки, диаграммы, формульные выражения, заготовленные векторные иллюстрации (клипарты), то есть в ней имеются средства, отдаленно напоминающие средства специализированных графических редакторов. Правда, среди этих средств нет ничего для создания и обработки растровых иллюстраций — их можно только импортировать из других программ, но зато есть средства для управления их визуализацией, например для изменения яркости, контрастности и масштаба изображения.

Несмотря на столь разностороннюю природу объектов, с которыми может работать текстовый процессор Word 2010, у них есть общие свойства, например такие, как размер, положение на странице, характер взаимодействия с текстом. Сначала мы остановимся на изучении самых общих свойств встроенных объектов, не обсуждая их природу, — это поможет освоить базовые приемы работы с объектами. А с конкретными свойствами конкретных объектов мы познакомимся чуть позже. Но перед тем как приступить к изучению приемов работы с объектами Word 2010, необходимо сделать важное замечание о целесообразности их применения. На этот счет существуют весьма противоречивые мнения.

1. Все объекты Microsoft Word 2010 безусловно можно использовать, если документ готовится для печати, то есть предполагается, что он будет передаваться заказчику или распространяться в виде бумажной копии, выполненной на принтере. Оформление документов с помощью встроенных объектов позволяет сделать их *представительными*.

2. Если документ предполагается передать в виде файла для последующей обработки (а именно так передают рукописи в редакции), то все собственные средства программы по созданию и размещению встроенных объектов не только бесполезны, но и вредны. Это связано с тем, что объекты Microsoft Word 2010 не стандартны и не поддерживаются профессиональными программами. Компания Microsoft имеет лидирующее положение в отрасли и может не считаться с общепринятыми стандартами и правилами, а внедрять свои. Поэтому объекты, созданные в программах этой компании, могут полноценно использоваться только в других программах той же компании.
3. Из последнего замечания вытекает еще одно направление для использования объектов, созданных в Microsoft Word. Их можно успешно экспортировать через буфер обмена Windows в другие программные продукты, входящие в пакет Microsoft Office 2010, например такие, как система управления электронными таблицами Excel, система управления базами данных Access и др.

Взаимодействие объектов Word с текстом и страницей

Управление размером и положением объекта. Взгляните на рис. 11.1. Здесь представлен графический объект, встроенный в текст документа. Этот объект обладает рядом свойств. Самое очевидное свойство — его размер. Когда объект выделен, вокруг него видны восемь квадратных маркеров. При наведении указателя мыши на один из маркеров указатель меняет форму и превращается в двунаправленную стрелку. В этот момент размер объекта можно менять методом протягивания мыши. Угловые маркеры позволяют пропорционально изменять размер объекта как по горизонтали, так и по вертикали. Четыре маркера, расположенные на сторонах воображаемого прямоугольника, позволяют управлять размером по одному направлению (по вертикали или горизонтали).

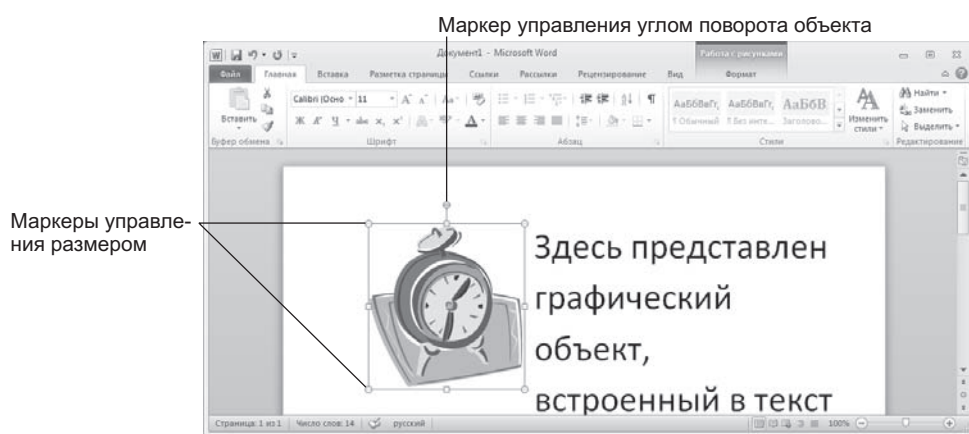


Рис. 11.1. Пример объекта, встроенного в текст

При наведении указателя мыши на границу окаймляющей объект рамки указатель меняет форму и превращается в четырехнаправленную стрелку. В таком

состоянии объект можно перетаскивать с помощью мыши по рабочему полю документа. Он займет новое положение в тот момент, когда левая кнопка мыши будет отпущена после перетаскивания.

Расширенное управление свойствами объектов. Вручную мы можем только управлять размером, поворотом и положением объекта на странице. Для управления всеми остальными свойствами объектов нужны дополнительные средства — их можно найти в двух местах:

- ☐ на вкладке, соответствующей типу объекта (она открывается автоматически, когда объект выделен);
- ☐ в диалоговом окне **Формат рисунка** (рис. 11.2), которое открывают из контекстного меню объекта (после щелчка правой кнопкой мыши на объекте).

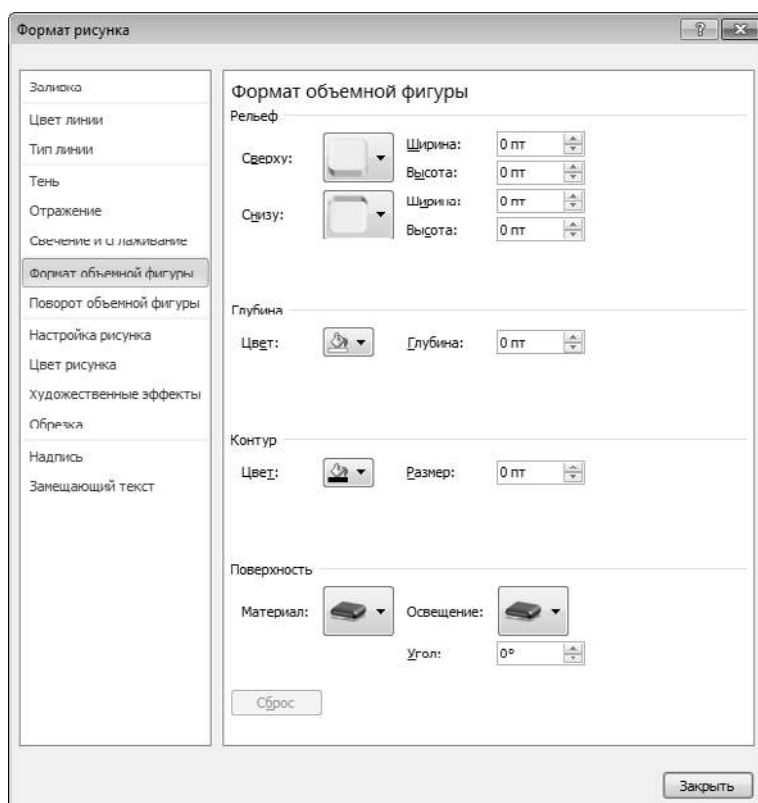


Рис. 11.2. Основное средство управления общими параметрами встроенного объекта

С помощью вкладок ленты управляют индивидуальными свойствами объектов (у разных типов объектов они различны), а с помощью диалогового окна **Формат рисунка** управляют наиболее общими свойствами, имеющимися у объектов любых типов.

Взаимодействие объекта с окружающим текстом. Вставив объект в текст, следует задать характер его взаимодействия с текстом. Средства для этого (кнопки

Положение и Обтекание текстом) представлены в группе параметров на вкладке Формат, в группе параметров Упорядочить. Возможны следующие варианты.

1. Вариант В тексте используют для графических объектов малого размера, сопоставимого с размерами символов текста. В этом случае объект вставляется в текстовую строку на правах графического символа и далее перемещается по странице только вместе с текстом.
2. Вариант Вокруг рамки предусматривает расположение текста вокруг воображаемой прямоугольной рамки, охватывающей весь контур объекта.
3. Вариант По контуру отличается от предыдущего тем, что воображаемая прямоугольная рамка не проводится и текст плавно обтекает контур объекта (если он криволинейный).
4. Вариант Сквозное — это прием обтекания, аналогичный обтеканию По контуру, но в данном случае текст обтекает объект не только снаружи, но и изнутри.
5. Вариант Сверху и Снизу оставляет справа и слева от рисунка пустые места, располагая текст, разрываемый объектом, лишь сверху и снизу объекта. Этот прием используют наиболее часто — его считают основным для объектов, ширина которых составляет более половины ширины страницы.
6. Вариант Перед текстом — это прием вставки объекта без обтекания. Текст и объект лежат на разных слоях, причем объект лежит выше и загораживает часть текста. Этим приемом пользуются, когда оформление важнее содержания.
7. Вариант За текстом — это еще один прием вставки объекта без обтекания. Текст и объект тоже лежат на разных слоях, но в данном случае объект лежит на нижнем слое и загорается текстом. Этот вариант используют для размещения текста на тематическом художественном фоне.

Прочие параметры взаимодействия объекта с окружающим текстом. Более тонкую настройку взаимодействия объектов с текстом выполняют с помощью элементов управления, имеющихся в диалоговом окне Разметка (Формат ► Упорядочить ► Обтекание текстом ► Дополнительные параметры разметки). В частности, здесь можно с помощью переключателей конкретно указать, с каких сторон объекта происходит обтекание, а с каких — нет. Здесь же можно указать величину интервала в миллиметрах между текстом и объектом.

Управление вертикальным и горизонтальным положением объекта относительно элементов печатной страницы. Завершив настройку взаимодействия объекта с текстом, приступают к размещению объекта на странице. Как уже говорилось выше, это можно сделать вручную методом перетаскивания объекта с помощью мыши, но более точную настройку выполняют с помощью командного меню кнопки Положение (Формат ► Упорядочить ► Положение). Здесь можно выбрать один из предустановленных вариантов расположения рисунка в тексте (рис. 11.3), или построить положение рисунка в тексте с помощью вкладки Положение диалогового окна Разметка.

К объекту, встроенному в текст, можно подходить с двух позиций: как к элементу оформления страницы или как к элементу оформления содержания, то есть текста. Разница заключается в том, что происходит с объектом во время редакци-

рования текста: он перемещается вместе с ним (с абзацами, к которым он при-
мыкает) или он неподвижен, а текст перемещается, обтекая объект по заданным
правилам.

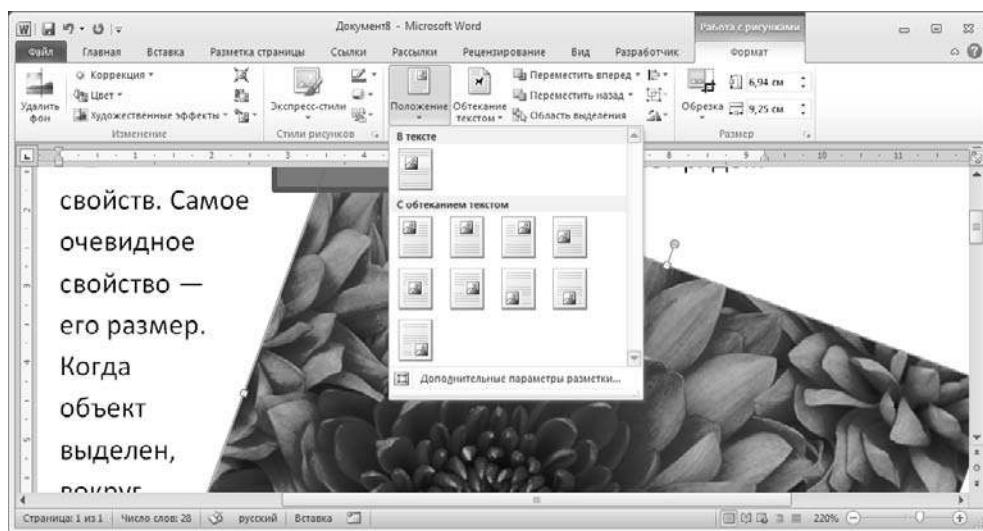


Рис. 11.3. Управление вертикальным и горизонтальным положением объекта

Вертикальное положение относительно текста задают установкой переключателя Положение и выбором объекта, относительно которого положение задается, например абзаца.

Чтобы объект был связан с элементом страницы и не перемещался вместе с текстом, на вкладке Положение диалогового окна Разметка устанавливают флажок Установить привязку. Чтобы объект мог перемещаться вместе с текстом, устанавливают флажок Перемещать вместе с текстом.

Управление свойствами объектов Microsoft Word

Управление размерами объекта. Мы знаем, что размерами встроенных объектов можно управлять перетаскиванием графических маркеров с помощью мыши. Это прием ручного управления. Однако существуют и приемы автоматического управления. Во-первых, размеры объекта можно задать, воспользовавшись элементами управления группы параметров Размер на вкладке Формат. Во-вторых, их реализуют с помощью элементов управления вкладки Размер рассмотренного выше диалогового окна Разметка (рис. 11.4). Счетчиками Высота, Ширина и Поворот задают вертикальные и горизонтальные размеры объекта, а также его угол поворота по часовой стрелке.

Размерами объектов можно управлять не только в абсолютном исчислении, но и в относительном (в процентах относительно исходного). Для этого служат счетчики группы Масштаб. Чтобы размеры объекта синхронно изменялись по вертикали и горизонтали, надо установить флажок Сохранить пропорции.

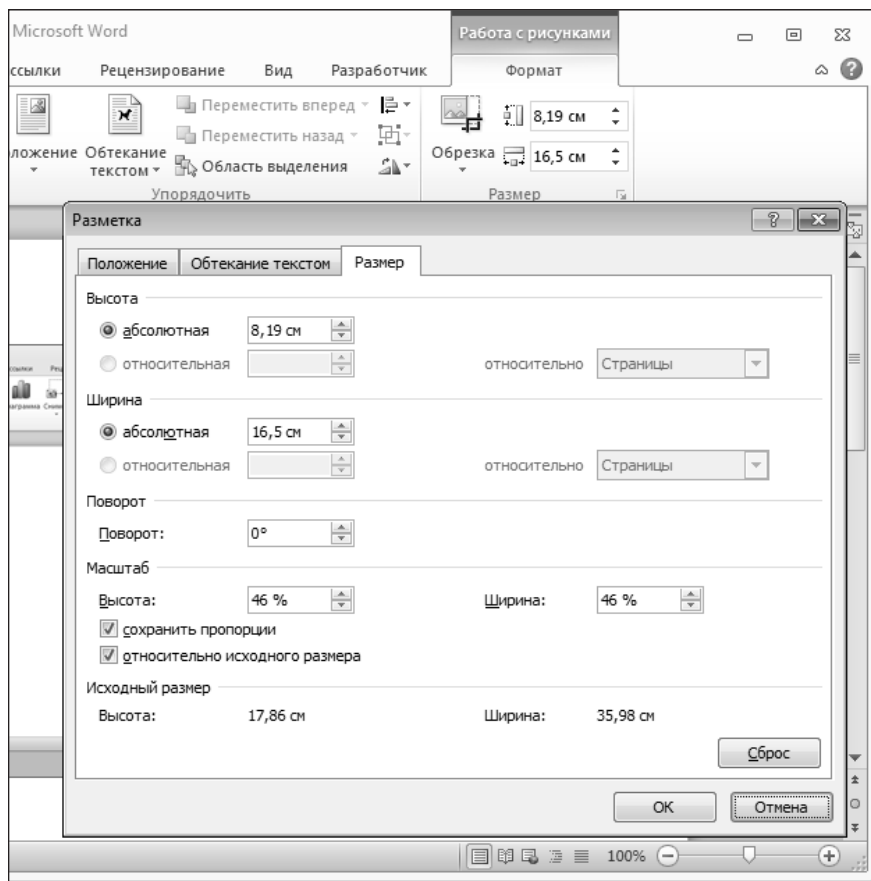


Рис. 11.4. Средства настройки размера объекта

Управление свойствами линии. Большинство объектов, создаваемых средствами самой программы Word 2010, имеют векторную природу, то есть в их основе лежат простейшие геометрические фигуры — линии. Эти линии, в свою очередь, имеют собственные свойства: толщину, цвет и тип. Управление этими свойствами выполняют либо с помощью инструментов группы параметров **Стили фигур** вкладки **Формат**, либо с помощью вкладок **Цвет линии** и **Тип линии** окна **Формат фигуры** (его можно вызвать по соответствующей команде контекстного меню фигуры).

Управление свойствами замкнутых линий. Замкнутые линии, в отличие от обычных, обладают дополнительным свойством — *заливкой*. Свойство заливки задают на вкладке **Заливка** окна **Формат объекта** (рис. 11.5.) или с помощью меню **Заливка фигуры** группы параметров **Стили фигур** на вкладке **Формат**. Заливка может быть *простой* и *комбинированной*.

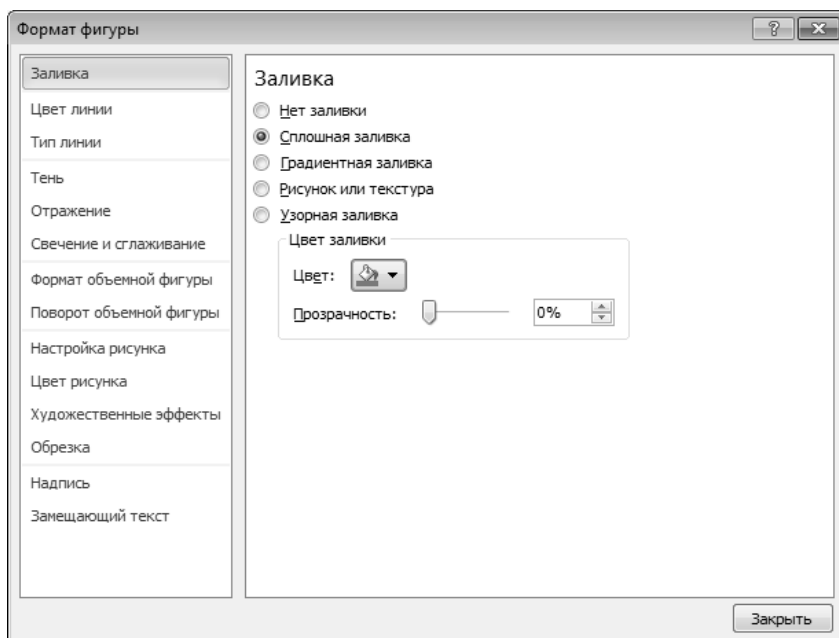


Рис. 11.5. Средства настройки заливки объекта

Простая заливка — одноцветная. Простой заливке при выборе варианта заливки соответствует переключатель *Сплошная заливка* на вкладке настройки параметров заливки окна *Формат фигуры*. Цвет заливки может быть одним из нескольких десятков стандартных, имеющих в палитре, или одним из дополнительных (выбирается в палитре с помощью кнопки *Другие цвета*).

Простая заливка, а также некоторые виды комбинированных заливок могут быть полупрозрачными, — тогда через закрашенные контуры может просвечивать текст или объект нижележащего слоя (рис. 11.6).

Комбинированная заливка имеет более сложный характер. В программе Word 2010 реализовано четыре метода комбинированной заливки:

- ☐ градиентная заливка;
- ☐ текстурная заливка;
- ☐ заливка рисунком (изображением-картой);
- ☐ заливка узором;

Градиентная заливка — это многоцветная заливка, при которой осуществляется плавный переход между заданными цветами. Количество исходных цветов, сами цвета и направление градиента произвольно выбираются на вкладке *Заливка* при установленном переключателе *Градиентная заливка*.

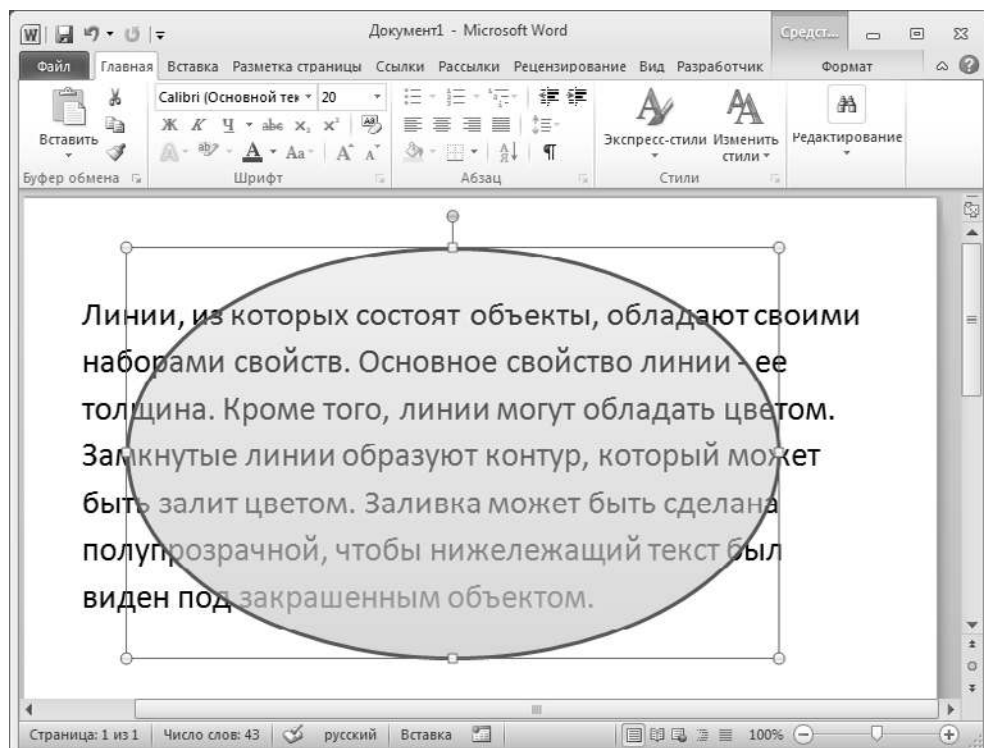


Рис. 11.6. Полупрозрачная заливка

Текстурная заливка — это заливка, воспроизводящая нерегулярную текстуру. Обычно используется для имитации поверхности материала. Выбор текстуры выполняют на вкладке **Заливка** окна **Формат фигуры** при переключателе вида заливки, установленном в положение **Рисунок** или **текстура** (рис 11.7).

Если представленных там текстур недостаточно, с помощью кнопок группы **Вставить из...** можно загрузить графический файл с изображением дополнительной текстуры (кнопка **Файл**), использовать текстуру, скопированную ранее в буфер обмена (кнопка **Буфер обмена**), или, загрузив картинку (кнопка **Картинка**), преобразовать ее в текстуру, установив переключатель **Преобразовать рисунок в текстуру**.

Заливка изображением-картой — это аналог текстурной заливки, при котором замкнутый контур заполняется специально подготовленным графическим изображением. Выбор изображения выполняют выбором файла, в котором оно хранится с помощью кнопки **Картинка** на вкладке **Заливка** при установленном переключателе вида заливки в **Рисунок** или **текстура**.

Заливка узором, как и заливка текстурой, — это заливка заранее подготовленным изображением, но имеющим регулярный характер. Выбор узора выполняют на вкладке **Заливка** при установленном переключателе вида заливки в **Узорная заливка**. Там же можно настроить цвет переднего плана рисунка узора и цвет его фона.

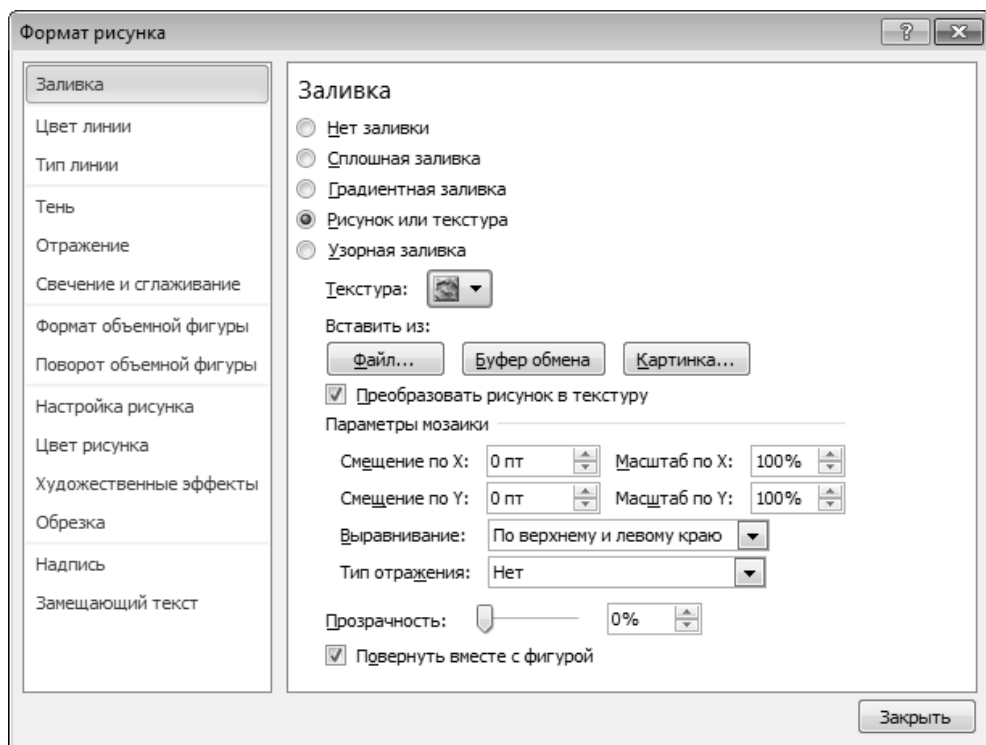


Рис. 11.7. Заливка узором

Взаимодействие объектов друг с другом

Мы рассмотрели, как происходит взаимодействие объектов с текстом и с элементами печатной страницы, но если на одной странице имеется несколько встроенных объектов, то они могут взаимодействовать и друг с другом. Характером этого взаимодействия тоже нужно управлять.

Первое, что нужно решить, — это разрешено ли объектам перекрывать друг друга. Для тех объектов, которым перекрывание разрешено, следует установить флажок **Разметка ► Положение ► Разрешить перекрывание**.

Напомним, что доступ к диалоговому окну **Разметка** открывается командой **Дополнительные параметры разметки** контекстного меню объекта. Так же это окно может быть вызвано командой вкладки **Формат: Упорядочить ► Положение ► Дополнительные параметры разметки**.

Управление взаимным положением объектов выполняют с помощью операций:

- ☐ группирования;
- ☐ задания порядка следования;
- ☐ выравнивания;
- ☐ распределения.

Группирование объектов. Если на странице представлено несколько объектов и при этом важно строго зафиксировать их взаимное расположение, то их объединяют в один комплексный (групповой) объект с помощью операции группирования. После этой операции свойства группового объекта можно настраивать точно так же, как мы настраивали свойства простейших объектов (рис. 11.8.), — ему может быть задан характер обтекания текстом, метод привязки к абзацу или к элементам печатной страницы и т. п.

Для группирования нескольких объектов их следует выделить (выделение нескольких объектов выполняют при нажатой клавише **SHIFT**), щелкнуть на любом из объектов группы правой кнопкой мыши и выбрать в контекстном меню команду **Группировать** ► **Группировать**. Сгруппированные объекты можно перемещать как единое целое. Чтобы разгруппировать объекты и получить доступ к индивидуальным свойствам каждого из них, надо выделить группу и дать команду **Группировать** ► **Разгруппировать**.

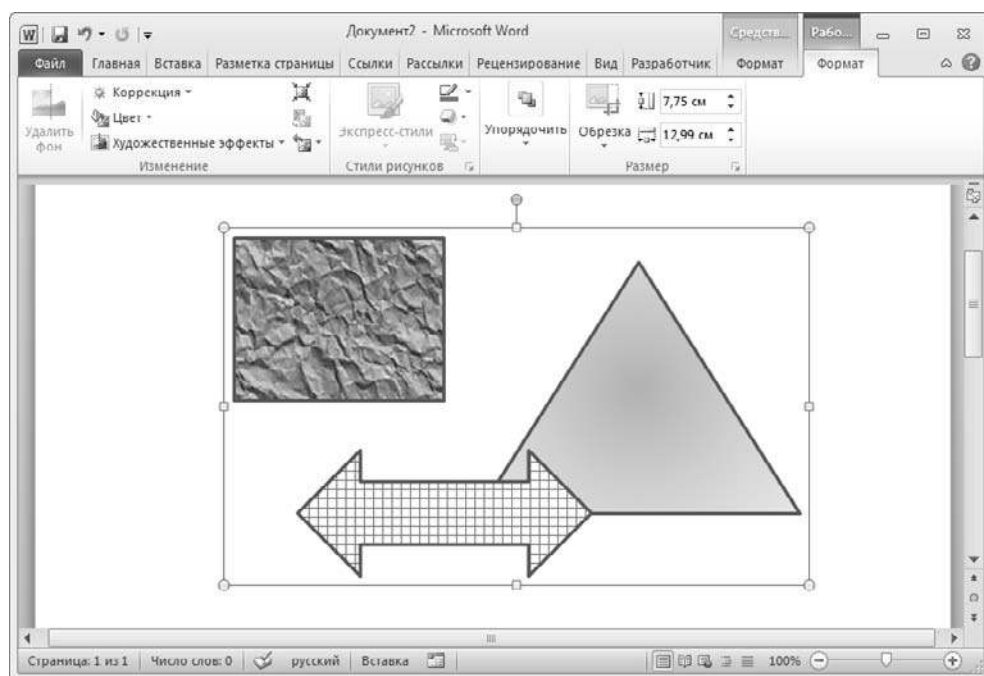


Рис. 11.8. Сгруппированный комплексный объект

Управление порядком следования объектов. Если на странице документа размещается несколько объектов, то предполагается, что у каждого объекта есть свой *слой*. По умолчанию порядок следования слоев связан с порядком создания объектов, то есть те объекты, которые были созданы раньше, лежат на слоях ниже, чем объекты, созданные позже. Если между объектами нет перекрытия, то мы не замечаем, что существует некий порядок следования объектов, однако, когда объекты перекрывают друг друга, этот порядок становится заметен.

Управляют порядком следования объектов с помощью команд **На передний план** и **На задний план** контекстного меню. Они открывают вложенные меню, средствами которых можно поднять объект на передний план, опустить на задний план, сместить на один слой вверх или вниз и задать положение объекта относительно текста.

Выравнивание объектов. Если объекты, составляющие композицию, не перекрывают друг друга, важно иметь средство их относительного выравнивания между собой. Выравнивание объектов выполняют до группирования, ведь после него объекты уже нельзя сдвинуть друг относительно друга. В этом случае операция группирования закрепляет взаимное расположение объектов. После нее объекты уже не могут сдвинуться друг относительно друга, и положением всей группы на странице можно управлять как единым целым. Чтобы выполнить выравнивание, необходимо воспользоваться инструментами группы **Упорядочить** вкладки **Формат** (рис. 11.9).

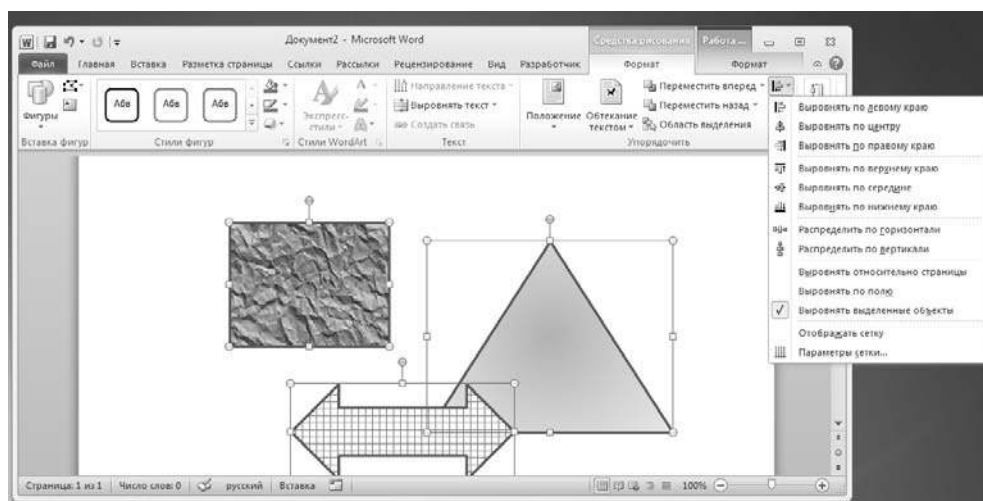


Рис. 11.9. Выравнивание и распределение объектов

Для выравнивания нескольких объектов между собой их следует выделить при нажатой клавише **SHIFT**, а затем дать команду **Формат** ► **Упорядочить** ► **Выровнять...** (и выбрать подходящий метод выравнивания).

Существует шесть методов выравнивания. Им соответствуют три команды горизонтального выравнивания (по левому краю, по правому краю, по центру) и три команды выравнивания вертикального (по верхнему краю, по нижнему краю, по середине). Следует обратить внимание на особенность действия команд выравнивания. Так, например, если два объекта выравниваются по *нижнему* полю, значит, они выравниваются по *нижнему* полю *нижнего* объекта. Выравнивание по *правому* полю — это выравнивание по *правому* полю самого *правого* объекта из числа выделенных и т. д. Если необходимо выполнить выравнивание относительно полей страницы, следует предварительно установить флажок меню **Формат** ► **Упорядочить** ► **Выровнять** ► **Выровнять относительно страницы**.

Распределение объектов. Эта операция родственна выравниванию. Ее суть в том, что между объектами устанавливаются равные интервалы по горизонтали или (и) вертикали. Соответственно, в меню команды **Формат** ► **Упорядочить** ► **Выровнять** имеются команды: **Распределить по горизонтали** и **Распределить по вертикали**.

Равномерное распределение объектов обычно выполняют после выравнивания, но, разумеется, до группирования. Нередко объекты выравнивают по вертикали и одновременно равномерно распределяют по горизонтали или, соответственно, наоборот. Дополнительное отличие команд распределения от команд выравнивания заключается еще и в том, что для взаимного выравнивания достаточно иметь два выделенных объекта, а для команд распределения должно быть выделено не менее трех объектов.

11.2. Ввод формул

Необходимость в наличии средства для ввода математических выражений в текстовый документ характерна для научно-технической документации. Одним из таких средств является специальное приложение Mathcad, представленное в главе 18. Но функции системы Mathcad намного шире, и есть немало оснований для того, чтобы иметь простое средство ввода формул в самом текстовом процессоре.

В программе Microsoft Word 2010 имеется встроенное средство редактирования формул. Оно позволяет создавать формульные объекты и вставлять их в текстовый документ. При необходимости вставленный объект можно редактировать непосредственно в поле документа.

Вставка формул в документ

Для вставки формулы в документ служит командная кнопка **Вставка** ► **Символы** ► **Формула**. Кнопка имеет меню, вызвав которое можно быстро вставить в документ шаблоны некоторых формул из коллекции формул. Если же нажать в основной области кнопки, будет открыта вкладка **Работа с формулами** ► **Конструктор**, а в документе появится поле формулы, данные в которое вводятся с помощью инструментов, размещенных на ленте (рис. 11.10).

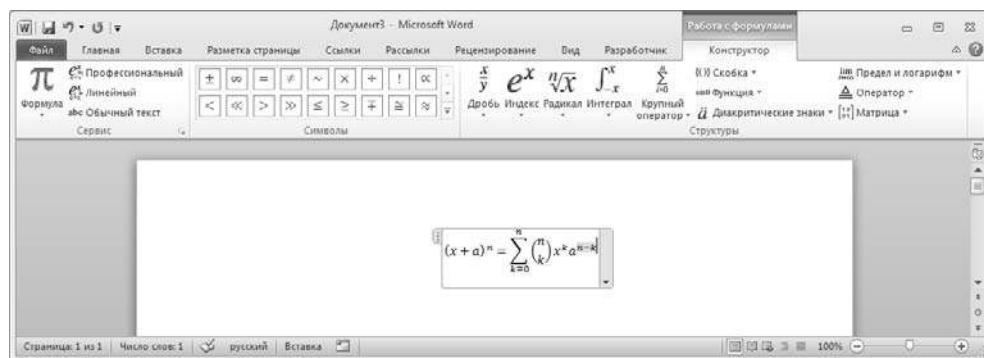


Рис. 11.10. Работа с формулами

Вкладка ленты Конструктор содержит несколько групп командных элементов. Группа Сервис содержит инструменты для настройки внешнего вида формул и ввода формул из так называемой коллекции формул, которую можно пополнять (Конструктор ► Сервис ► Формула ► Сохранить выделенный фрагмент в коллекцию формул). В правом нижнем углу группы Сервис находится кнопка Параметры формул, нажав на которую можно вызвать одноименное окно, содержащие общие параметры настройки формул.

Группа Символы предназначена для быстрого ввода различных специальных символов, используемых в формулах.

Группа Структуры позволяет добавлять в поле редактирования своеобразные шаблоны для построения сложных формул.

Ввод и редактирование формул завершается щелчком левой кнопкой мыши где-либо в поле документа вне области ввода формулы. Введенная формула автоматически вставляется в текст в качестве объекта. Далее ее можно переместить в любое иное место документа через буфер обмена (CTRL+X — вырезать; CTRL+V — вставить). Для редактирования формулы непосредственно в документе достаточно выполнить на ней двойной щелчок. При этом автоматически открывается вкладка Конструктор.

При работе с редактором формул следует стремиться к максимальной полноте вводимых выражений. Так, например, выражение (формула) может содержать компоненты, ввод которых возможен и без использования редактора формул, но для удобства работы и простоты дальнейшего редактирования следует вводить всю формулу целиком только в редакторе формул, не используя иные средства.

$$S = at^2/2 \text{ — неправильно; } S = \frac{at^2}{2} \text{ — правильно.}$$

11.3. Работа с таблицами

Данные, представленные в табличной форме, отличаются наглядностью. Таблицы всегда были неотъемлемым атрибутом печатной научно-технической документации, а в последние годы являлись и эффективным средством оформления веб-страниц Интернета. Многие веб-дизайнеры используют таблицы (в том числе и скрытые), чтобы принудительно управлять отображением данных на экране клиента и не доверять этот ответственный процесс браузеру. Так, например, таблицы — это простейшее средство для имитации на веб-странице газетного или журнального текста, имеющего две и более колонок.

Ячейки таблиц могут содержать не только текст, но и графические и прочие объекты. Благодаря этому можно размещать несколько иллюстраций по ширине веб-страницы.

Надо отметить, что в наши дни верстка веб-страниц с использованием таблиц уступает место более современному подходу. Такой подход к разработке дизайна веб-страниц предусматривает использование специальных тегов (так называемых DIV-тегов), которые позволяют создавать произвольные блоки информации и гибко управлять их взаимным расположением и другими свойствами.

При создании страниц с использованием таблиц можно управлять методом представления ячеек и рамок, как внешних, так и внутренних. При создании печатных документов таблицы оформляют так, чтобы они соответствовали стилю и содержанию документа. При создании веб-страниц существует прием, когда рамки вообще не отображают, а между ячейками делают зазор. В результате этого объекты, находящиеся в ячейках, образуют ровные регулярные структуры на экране, в то время как никаких следов таблиц на экране не видно.

Текстовый процессор Microsoft Word обладает удивительно гибкими и мощными средствами создания таблиц как для печатных, так и для электронных документов.

Для создания таблиц в Microsoft Word применяют инструменты, находящиеся по адресу Вставка ► Таблицы ► Таблица (рис. 11.11).

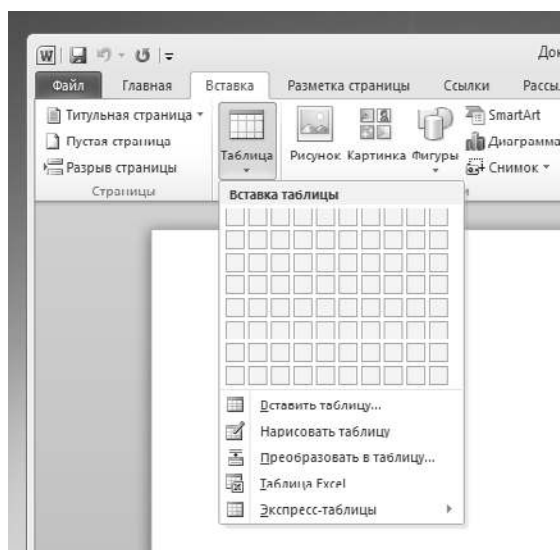


Рис. 11.11. Инструменты для добавления таблиц в документы

Для того чтобы быстро вставить в документ таблицу, можно выделить нужное количество ячеек таблицы (до 10 в ширину и 8 в высоту), которые расположены в верхней части меню кнопки Таблица.

Если есть необходимость вставить в документ таблицу произвольного размера, настроив при этом ее параметры, нужно воспользоваться командой меню Вставить таблицу.

Таблицу можно нарисовать «от руки», применив команду меню Нарисовать таблицу.

Кроме того, система содержит набор предварительно подготовленных стилей таблиц (пункт меню Экспресс-таблицы), которые позволяют добавлять в документ таблицы, отформатированные различными способами.

Для работы с таблицами служат специальные вкладки Работа с таблицами ► Конструктор и Работа с таблицами ► Макет.

Создание таблиц

Ячейки таблицы, доступные в верхней части меню кнопки **Вставка** ▶ **Таблицы** ▶ **Таблица**, используют для создания простейших таблиц небольшого размера. Созданные таким методом таблицы можно в дальнейшем развивать, редактируя таблицу.

Команду **Вставка** ▶ **Таблицы** ▶ **Таблица** ▶ **Вставка таблицы** используют для создания более сложных таблиц. Она открывает диалоговое окно **Вставка таблицы** (рис. 11.12), в котором задают число строк и столбцов, а также ширину столбцов.

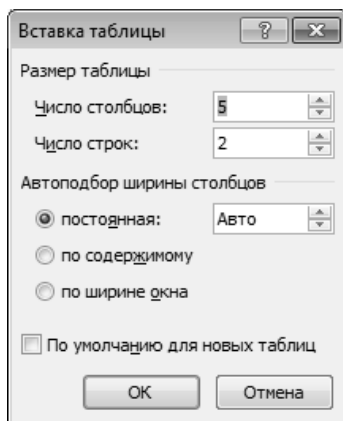


Рис. 11.12. Окно Вставка таблицы

Режим автоподбора задают соответствующим переключателем:

- ❑ **постоянная** — общая ширина таблицы в режиме **Авто** равна ширине поля набора документа, а ширина каждого столбца постоянна и зависит от количества столбцов (режим удобен при создании печатных документов);
- ❑ **постоянная** — при вводе конкретного значения в поле задания ширины столбца редактор создаст таблицу, столбцы которой будут обладать именно такой шириной. Недостаток этого режима заключается в том, что общий размер таблицы может превысить размер печатной области страницы, что потребует дополнительных настроек;
- ❑ **по содержимому** — ширина каждого столбца пропорциональна объему данных, содержащихся в нем (режим удобен при создании электронных документов, распространяемых в формате текстового процессора);
- ❑ **по ширине окна** — специальный режим для таблиц, размещаемых на веб-страницах (окончательное форматирование таблицы происходит не в момент ее создания, а во время просмотра).

Таблицы сложной структуры удобно создавать методом «рисования». Порядок действий, необходимых для создания таблиц этим методом, рассмотрен в упражнении 11.1.

Редактирование таблиц

Говоря о редактировании таблиц, мы имеем в виду не редактирование их содержимого, а только редактирование их структуры. Редактирование содержимого осуществляется обычными средствами, рассмотренными в предыдущей главе. Фактически редактирование структуры таблиц сводится к следующим операциям:

- ☐ добавление заданного количества строк;
- ☐ добавление заданного количества столбцов;
- ☐ удаление выделенных ячеек, строк и столбцов;
- ☐ слияние выделенных ячеек;
- ☐ разбиение выделенных ячеек.
- ☐ изменение размеров выделенных ячеек.

Комбинируя вышеуказанные операции, можно на базе таблиц с простой структурой готовить таблицы, имеющие сложную структуру.

Средства для выполнения этих операций находятся среди инструментов вкладок Макет и Конструктор (рис. 11.13).

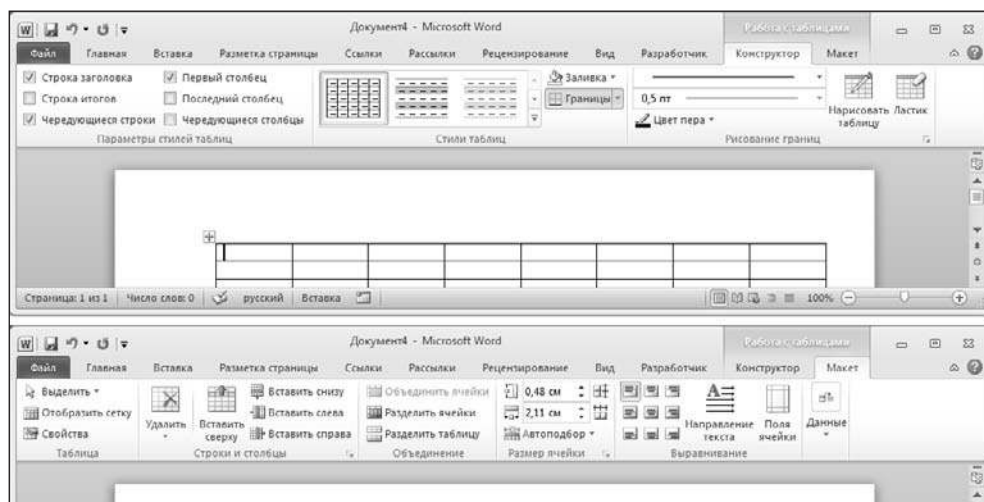


Рис. 11.13. Инструменты Microsoft Word для редактирования таблиц

Используя инструменты вкладки Конструктор, можно, во-первых, настраивать внешний вид таблицы, управляя границами и заливкой ячеек с применением стилевого оформления или ручной настройки параметров. Во-вторых, применяя инструменты группы Рисование границ, можно «дорисовывать» или «стирать» элементы таблицы.

Вкладка Макет ориентирована в основном на вставку и удаление блоков ячеек (группа Строки и столбцы), на настройку объединения и разделения ячеек (Объединение), на настройку размеров ячеек (Размер ячейки). Также с помощью ин-

струментов этой вкладки можно задавать выравнивание и направление текста в ячейках таблицы.

Многие действия можно произвести, используя контекстное меню выделенной области таблицы.

Форматирование таблиц

При работе с таблицами следует различать *форматирование таблиц* и *форматирование содержимого*. В первом случае происходит управление размерами структурных элементов таблицы (ячеек, строк, столбцов и т. п.), а во втором — управление размещением содержимого ячеек.

Форматирование таблиц можно выполнять в командном или интерактивном режиме. В командном режиме для этой цели используют диалоговое окно Свойства таблицы (Макет ► Таблица ► Свойства) (рис. 11.14).

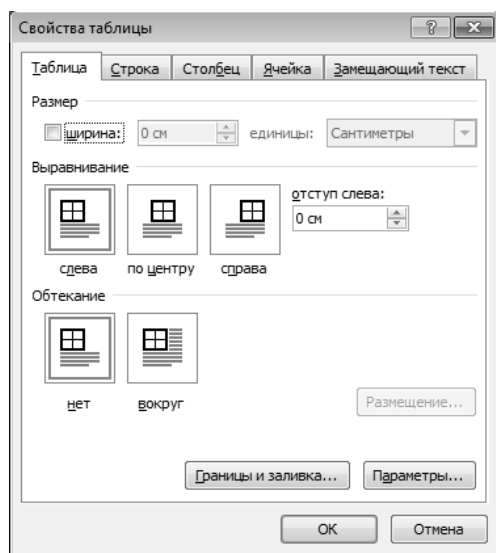


Рис. 11.14. Окно Свойства таблицы

Его можно открыть и из контекстного меню таблицы, если щелкнуть в ее пределах правой кнопкой мыши. Элементы управления вкладок диалогового окна Свойства таблицы позволяют:

- ❑ задать метод выравнивания таблицы относительно страницы документа (Свойства таблицы ► Таблица ► Выравнивание);
- ❑ задать метод взаимодействия таблицы с окружающим текстом (Свойства таблицы ► Таблица ► Обтекание);
- ❑ определить или переопределить вариант оформления внешних и внутренних рамок таблицы, а также настроить характер оформления ячеек (Свойства таблицы ► Таблица ► Границы и заливка);

- ❑ задать размеры внутренних полей в ячейках и интервалы между ячейками (Свойства таблицы ► Таблица ► Параметры);
- ❑ назначить параметры текущей строки или выделенных строк (Свойства таблицы ► Строка);
- ❑ назначить параметры текущего столбца или выделенных столбцов (Свойства таблицы ► Столбец);
- ❑ назначить параметры текущей ячейки или выделенных ячеек (Свойства таблицы ► Ячейка).
- ❑ задать параметры текста, замещающего таблицу, размещенную на веб-странице, при ее загрузке браузером или в других случаях (Свойства таблицы ► Замещающий текст).

В интерактивном режиме таблицу форматируют с помощью маркеров, появляющихся при наведении указателя мыши на таблицу или ее элементы. Маркер в левом верхнем углу таблицы позволяет перемещать таблицу по рабочему полю документа. Маркер в правом нижнем углу позволяет управлять общими размерами таблицы. Маркеры изменения размера, появляющиеся при наведении указателя мыши на рамки таблицы, позволяют интерактивно изменять размеры столбцов и строк методом перетаскивания.

Ввод и форматирование содержимого таблиц

Выделение нужной ячейки для ввода текста выполняют с помощью мыши. Отдельную ячейку выделяют тройным щелчком левой кнопки. Перемещение между ячейками выполняют клавишей TAB (к следующей ячейке) или комбинацией SHIFT+TAB (к предыдущей ячейке). Для навигации по ячейкам таблицы можно также использовать клавиши управления курсором. В тексте курсорные клавиши выполняют перемещение курсора внутри ячейки, но по достижении границы текста они позволяют переходить к соседним ячейкам.

Все команды форматирования текста относятся к выделенному элементу. Выделенным элементом может быть любая ячейка, строка (группа строк), столбец (группа столбцов) или вся таблица в целом. Группы ячеек выделяют методом протягивания мыши. Большинство команд, связанных с форматированием элементов таблицы и содержащихся в них объектов, можно выполнить с помощью вкладки Главная.

Автоматическое форматирование таблиц

Автоматическое форматирование таблиц выполняют с помощью инструментов группы Стили таблиц, доступного на вкладке Конструктор. Набор предлагаемых форматов представлен в списке Стили таблиц. Результат, получающийся при их использовании, можно просмотреть на формируемой таблице, наводя указатель на соответствующие миниатюры стилей оформления. Для применения к таблице выбранного стиля следует подтвердить его щелчком левой кнопки мыши.

После применения стиля к таблице можно приступить к более тонкой ее настройке с помощью вышеописанных «ручных» методов. При необходимости список встроенных стилей можно пополнить собственными стилями.

11.4. Работа с диаграммами

Диаграммы являются удобным средством визуального представления данных и наряду с таблицами очень широко используются в научно-технической документации. Для создания диаграмм текстовый процессор *Microsoft Word* имеет встроенное средство, основанное на использовании *Microsoft Office Excel 2010*. Если Excel не установлен, применяется средство *Microsoft Graph*.

Текстовый процессор *Microsoft Word 2010* предоставляет два метода для вставки диаграмм в документ. Более общий метод основан на том, что сначала в документ вставляется некая произвольная диаграмма, с которой связана некая произвольная *базовая таблица* данных. Далее производится настройка диаграммы, которая состоит в настройке внешнего вида и в редактировании содержания. Поскольку содержание основано на базовой таблице, то оно редактируется путем заполнения этой таблицы нужными данными.

Второй, частный метод основан на том, что диаграмма создается на базе конкретной таблицы, имеющейся в документе. В этом случае настройка диаграммы состоит только в настройке внешнего вида. Этот метод очевидно более удобен, но злоупотреблять им не следует, поскольку данные в таблице и диаграмме дублируют друг друга, а не во всяком документе это оправдано. Приемы создания диаграмм на базе таблиц документа мы рассмотрим в упражнении 11.2.

Создание базовой диаграммы

Создание диаграммы начинается с создания базовой диаграммы командой **Вставка** ► **Иллюстрации** ► **Диаграмма**. В открывшемся диалоговом окне (рис. 11.15) следует выбрать вид вставляемой диаграммы.

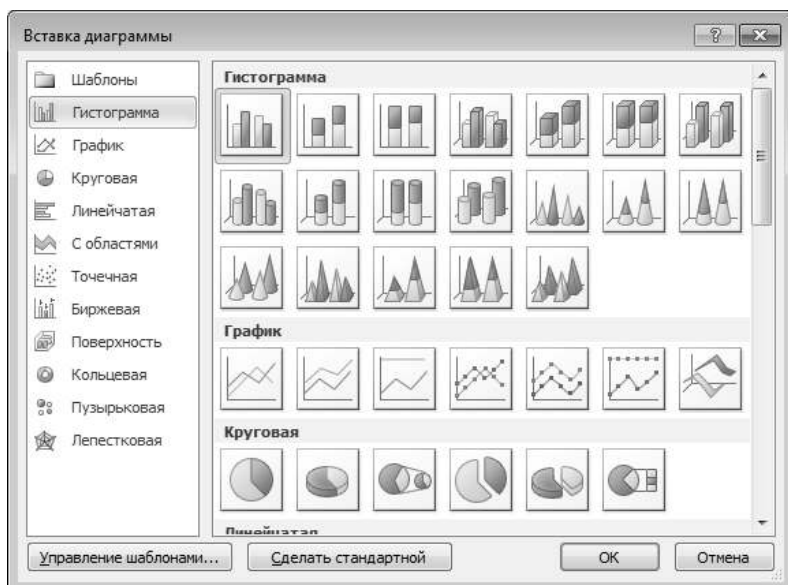


Рис. 11.15. Вставка новой диаграммы

После выбора типа диаграммы и нажатия на кнопку ОК в документ вставляется диаграмма, с которой связана некая *базовая таблица* (рис. 11.16), открытая для редактирования в Microsoft Excel. Рассматривайте эту таблицу как шаблон. Ее ячейки следует заполнить собственными данными, причем заполнение можно автоматизировать путем импорта данных из какой-либо иной таблицы, например из таблицы Microsoft Excel, хранящейся в отдельном файле, или из таблицы, которая находится в документе Microsoft Word.

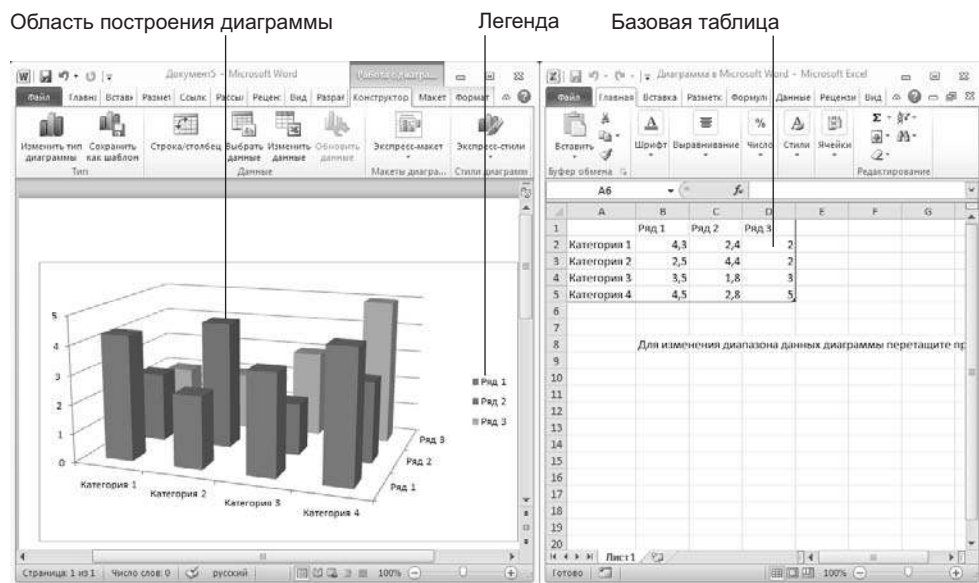


Рис. 11.16. Сначала в документ вставляется произвольная диаграмма и связанная с ней таблица. Далее диаграмма и таблица редактируются по месту

После того как работа по вводу данных в таблицу завершена, следует закрыть окно Microsoft Excel. Для того чтобы продолжить редактирование данных, можно воспользоваться пунктом **Изменить данные** контекстного меню диаграммы.

Настройка внешнего вида диаграммы

Существует множество различных типов диаграмм и графиков, отличающихся способом визуального представления связанных с ними данных.

Настройка параметров диаграммы, вставленной в документ, производится средствами вкладок **Конструктор**, **Макет** и **Формат** группы **Работа с диаграммами**, которая появляется на ленте при выделении диаграммы.

Выбор типа диаграммы производят в диалоговом окне **Тип диаграммы** (**Конструктор** ► **Изменить тип диаграммы**). Выбрав форму диаграммы, приступают к ее настройке. Настройка диаграммы состоит в выборе *элементов оформления диаграммы* и *элементов представления данных* и выполняется в диалоговых окнах, вызываемых из контекстного меню соответствующих элементов или средствами вкладок группы **Работа с диаграммами**.

Элементы представления данных — это точки на графиках, столбцы гистограмм, секторы круговых диаграмм — в общем, все то, что служит для непосредственного отображения данных. *Элементы оформления* — это название диаграммы, названия ее осей, «легенда» (специальное поле, в котором приведены условные обозначения для групп элементов данных), подписи к элементам данных и линии координатной сетки. Настройку выполняют подключением или отключением тех или иных элементов.

Элементы диаграммы бывают *связанными* или *присоединенными*. Так, например, название диаграммы, названия ее осей и легенду можно редактировать отдельно — это *присоединенные элементы оформления*. Подписи к элементам данных редактировать на диаграмме нельзя — они связаны со значениями в базовой таблице и потому считаются *связанными элементами*.

Для каждого из присоединенных элементов оформления можно выполнить индивидуальное форматирование. Для этого надо в поле диаграммы щелкнуть дважды на поле присоединенного элемента — откроется соответствующее диалоговое окно форматирования (Формат легенды, Формат оси, Формат ряда данных, Формат области диаграммы и т. д.). Состав вкладок и других элементов управления этих диалоговых окон зависит от свойств конкретного присоединенного элемента. Так, например, средства форматирования осей диаграммы отличаются от средств форматирования легенды.

Настройка элементов данных и элементов оформления — это как бы внутренние средства настройки диаграмм. Они определяют свойства диаграммы как объекта. Однако возможно также и редактирование объекта в целом в составе документа. Так, например, для выделенной диаграммы можно с помощью мыши изменять горизонтальный и вертикальный размеры объекта путем перетаскивания маркеров. При изменении размера диаграммы возможно автоматическое перемасштабирование ее элементов оформления.

11.5. Работа с графическими объектами

В документах *Microsoft Word* можно использовать два типа графических объектов: *фигуры* и *изображения*. На русском языке разница между этими терминами не очевидна, и мы поясним, что под ними понимается в текстовом процессоре *Word*. *Фигуры* — объекты векторной природы (линии, прямые и кривые, геометрические фигуры, стандартные и нестандартные). Простейшие средства для их создания есть в самом текстовом процессоре.

Изображения — растровые объекты. Текстовый процессор не имеет средств для их создания, поэтому они вставляются как внешние объекты из файла, подготовленного другими средствами (графическим редактором, с помощью сканера, цифровой камеры, графического планшета).

Фигуры всегда внедрены в документ — их можно редактировать непосредственно по месту. Изображения вставляют в документ методом связывания или внедрения. Их редактирование средствами текстового процессора возможно, но только в ограниченных пределах.

Для добавления графических объектов в документ служит группа параметров вкладки *Иллюстрации* вкладки *Вставка*. Так, для вставки векторных фигур, которые

можно редактировать средствами Microsoft Word, используется кнопка **Фигуры**. Для вставки растровых изображений применяется кнопка **Рисунок**. Кнопка **Картинка** используется для работы с наборами изображений (клипартами). С ее помощью в документ можно вставить не только растровые изображения, но и видеоклипы.

Работа с фигурами

Создание и редактирование фигур. Для добавления в документ векторного рисунка служит кнопка **Вставка** ► **Иллюстрации** ► **Фигуры**. В выпадающем меню кнопки следует выбрать подходящую фигуру (рис. 11.17). В его категориях представлены заготовки для создания линий, прямых и кривых, простейших геометрических фигур, фигурных стрелок и выносных линий, чертежных элементов для блок-схем и функциональных схем и пр.

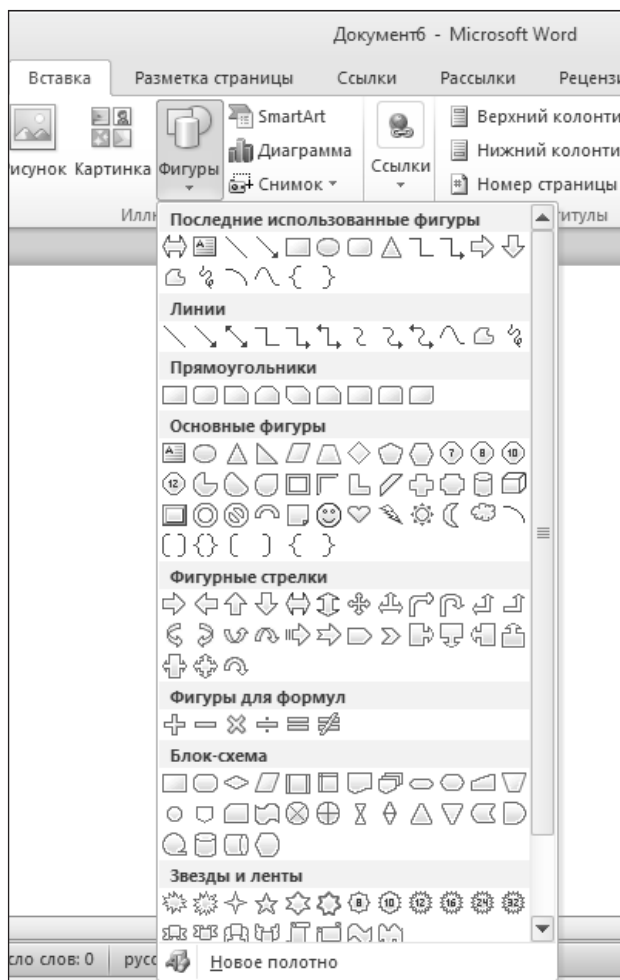


Рис. 11.17. Добавление векторной иллюстрации в документ

После выбора фигуры следует с помощью мыши (с зажатой левой кнопкой) «нарисовать» фигуру в том месте документа, куда она должна быть внедрена.

Для настройки фигуры могут использоваться обычные средства редактирования объектов. В частности, это настройка размеров и поворота фигуры с помощью мыши, настройка параметров фигуры с помощью окна свойств, вызываемого из ее контекстного меню. Параметры векторного изображения можно настраивать с использованием средств вкладки **Формат** группы **Средства рисования**, которая становится доступной при выделении фигуры (рис. 11.18).

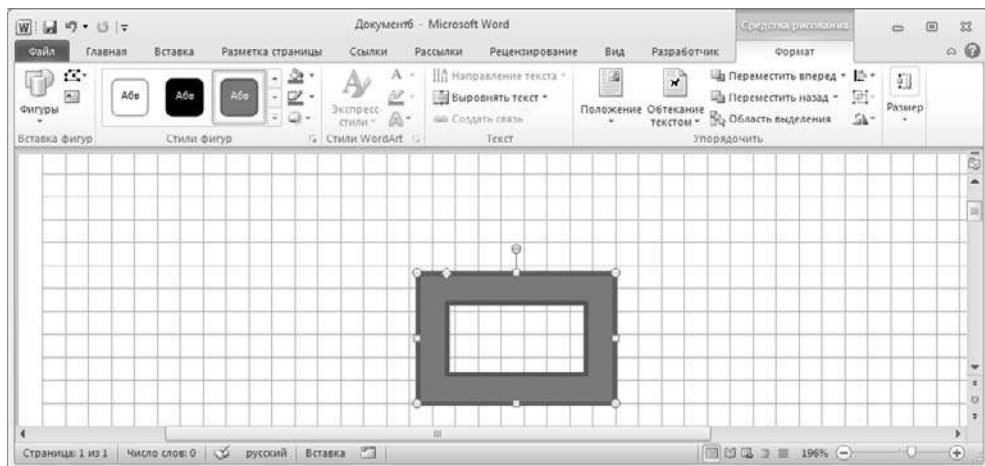


Рис. 11.18. Фигура, добавленная в документ

При создании и редактировании векторных объектов используют следующие приемы и средства.

1. Векторные объекты создают путем их выбора из категорий списка **Фигуры**.
2. Их размер редактируют путем перетаскивания маркеров выделенного объекта в поле документа.
3. Удобным средством, упрощающим создание геометрических фигур, является вспомогательная координатная сетка. Командой **Вид** ► **Сетка** можно включить отображение сетки. Новые объекты будут по умолчанию привязаны к сетке, их позиционирование и изменение размеров будет осуществляться в соответствии с шагом сетки. Этот режим удобен, если создаются простые (преимущественно прямолинейные) геометрические фигуры. При редактировании готовых фигур привязка к узлам сетки может создавать неудобства — в этом случае сетку отключают или выполняют перемещение объектов при нажатой клавише **ALT**.
4. Толщина контурной линии и цвет заливки объекта относятся к свойствам объекта. Все свойства объектов можно редактировать с помощью вкладки **Формат** группы **Средства рисования**, а также в диалоговых окнах **Формат фигуры** и **Разметка**. Диалоговые окна можно вызвать либо с помощью кнопок, размещенных на вкладке **Формат**, либо через контекстное меню объекта с помощью команд **Формат фигуры** и **Дополнительные параметры разметки**.

5. Поворотом объекта можно управлять. Для интерактивного управления поворотом можно воспользоваться маркером поворота, который обычно расположен в верхней части только что добавленного в документ объекта и имеет зеленый цвет. Параметры поворота можно задать на вкладке **Размер** диалогового окна **Разметка**.
6. Для того чтобы в интерактивном режиме изменить некоторые параметры фигуры, следует воспользоваться желтым ромбовидным маркером, который присутствует у фигур со сложной структурой.
7. Взаимодействие рисованного объекта с окружающим текстом может быть достаточно сложным. Так, например, текст может обтекать рисунок по заданной схеме, но он может лежать и поверх рисунка, и под ним. Выбор метода взаимодействия рисунка с текстом выполняют на вкладках **Положение** и **Обтекание** текстом в диалоговом окне **Разметка**.

Создание надписей в поле рисунка. Рисованные объекты могут содержать текстовые элементы, например заголовки, буквенные или цифровые обозначения на схемах и чертежах. В принципе, необходимые надписи можно создать и основными средствами текстового процессора, но в этом случае очень трудно обеспечить точное положение рисунка относительно связанного с ним текста, особенно если текст не окончателен и может далее редактироваться и форматироваться. Для веб-страниц этот метод вообще неприемлем, поскольку они формируются при каждом просмотре, причем непредсказуемым образом.

Для создания текстовых элементов, присоединенных к фигурам или рисункам, служит специальное средство **Надпись** (**Вставка** ► **Фигуры** ► **Основные фигуры** ► **Надпись**). Создав фигуру, рядом создают элемент **Надпись**. В поле надписи вводят необходимый текст, после чего надпись можно редактировать. Ее размер подгоняют под размер содержащегося в ней текста перетаскиванием маркеров. Прочие свойства надписи задают с помощью вкладки **Формат** группы **Средства рисования**, с помощью диалоговых окон **Формат фигуры** и **Разметка**, а также, в части форматирования шрифта надписи, с помощью стандартных инструментов вкладки **Главная**.

С точки зрения настройки параметров фигуры, надпись ничем не отличается от стандартных фигур. С точки зрения форматирования текста надписи, можно пользоваться обычными средствами форматирования.

Создав объект **Надпись**, его можно сгруппировать с рисунком, и тогда они будут представлять цельную композицию.

Для фигур есть особое средство создания текстового оформления — текст может размещаться в поле фигуры. Это выполняют командой **Добавить текст** в контекстном меню фигуры. Если текст слишком велик, можно либо изменить размер автофигуры путем перетаскивания ее маркеров, либо изменить формат текста, уменьшив размер шрифта средствами вкладки **Главная**. Этот прием используют при создании блок-схем и функциональных схем устройств.

Работа с клипартами

Создание достаточно сложных композиций может быть очень трудоемким. В таких случаях используют готовые библиотеки (*коллекции*) рисунков (*клипартов*), в том числе и тематических. Такие библиотеки распространяются на отдельных компакт-дисках, их можно найти в Интернете, но базовая, простейшая коллекция может быть установлена вместе с текстовым процессором — она входит в комплект поставки пакета *Microsoft Office*.

Для вставки клипартов используют команду Вставка ► Иллюстрации ► Картинка. При этом открывается Область задач в режиме Картинка. Это название достаточно условное, поскольку клипарт — понятие расширенное. К клипартам относят не только графические объекты, но и звуковые клипы и видеоклипы — их тоже можно вставить в документ с помощью этого средства.

Для поиска графических клипартов раскройте список Искать объекты и оставьте флажки только в нужных категориях. Затем вы можете сразу щелкнуть на кнопке Найти. На панели появятся изображения всех найденных клипартов. Поиск можно ограничить ключевыми словами, введенными в поле Искать (рис. 11.19).

Разыскав нужный клипарт, его можно вставить в документ простым щелчком. При необходимости можно вызвать контекстное меню изображения клипарта и, например, воспользовавшись командой Копировать, скопировать его в буфер обмена для последующей вставки.

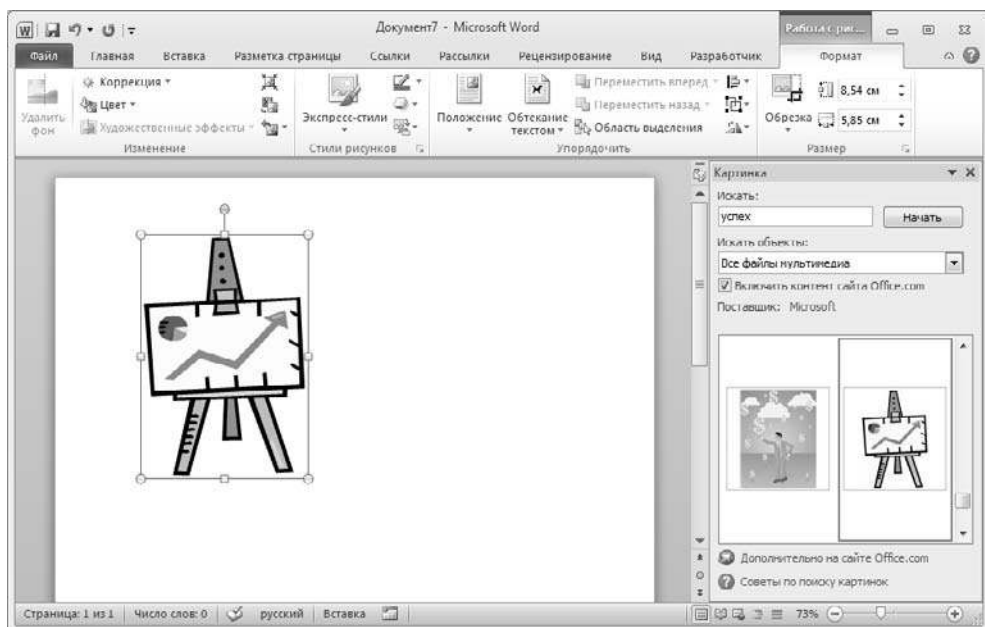


Рис. 11.19. Изображение, вставленное в документ

При работе с клипартами следует иметь в виду, что подобрать именно тот клипарт, который наилучшим образом соответствует характеру документа, можно

далеко не всегда. Поэтому клипарты следует рассматривать не как готовые средства оформления, а как заготовки для их создания. Клипарты — это композиционные объекты. Их можно «разбирать» на составляющие, редактировать их элементы по отдельности, создавать композиции из объектов, взятых из разных клипартов. Все это выполняется путем редактирования клипартов, вставленных в документ.

Обычный порядок редактирования клипартов следующий:

- ☐ клипарт выделяют щелчком левой кнопки мыши;
- ☐ открывают его контекстное меню щелчком правой кнопки;
- ☐ в контекстном меню выбирают команду **Изменение рисунка** — включается режим редактирования;
- ☐ в этом режиме работают с отдельными объектами, составляющими рисунок.

При работе с объектами клипарта используют команды разгруппировки и изменения порядка. Если из сложной композиции надо выделить один составляющий объект, то простейший прием состоит не в том, чтобы выделить все элементы, которые в него входят, а в том, чтобы удалить те, которые в него не входят. После каждого из удалений можно подавать отменяющую команду **CTRL+Z**, проверяя, что изменилось в составе рисунка. Если изменения желательны, их восстанавливают командой **CTRL+Y**, а если нет — переходят к выбору и удалению других элементов.

Комбинирование объектов, принадлежащих разным клипартам, выполняют путем копирования через буфер обмена *Windows* (**CTRL+C** и **CTRL+V**). При создании новых объектов из готовых клипартов часто приходится изменять размер итогового рисунка. Простейший способ для этого — с помощью мыши перетаскивать маркеры изменения размера объекта.

Специальные средства оформления. Эти средства оформления представлены кнопками на вкладке **Формат**, также ими можно воспользоваться из диалогового окна **Формат фигуры**:

- ☐ управлять цветом заливки, цветом контура и цветом текста;
- ☐ управлять толщиной сплошных линий и параметрами штриха для штриховых линий;
- ☐ преобразовывать линии в стрелки и управлять формой их концов;
- ☐ создавать теневые эффекты;
- ☐ создавать трехмерные эффекты.

Если к объекту применен теневой или трехмерный эффект, то редактировать результат этого эффекта непосредственно в поле документа нельзя, поскольку в отличие от контуров плоских объектов контуры трехмерных эффектов не являются объектами и не имеют управляющих маркеров. Поэтому для объектов, имеющих теневое или трехмерное оформление, используют иные приемы редактирования:

- ☐ выделяют объект в поле документа;
- ☐ используют элементы управления, расположенные в группе **Формат** ► **Стили фигур** или соответствующие инструменты окна **Формат фигуры** для настройки параметров эффекта;
- ☐ наблюдая результаты применения эффекта, вносят необходимые коррективы.

Работа с изображениями

Под *изображениями* понимаются растровые графические объекты, исполненные посторонними программными средствами или полученные из внешнего источника. Они вставляются в документ методом связывания или внедрения. Общая команда для вставки таких объектов — **Вставка ► Иллюстрации ► Рисунок**. По этой команде открывается стандартное диалоговое окно **Добавление рисунка**, в котором и производится выбор файла, содержащего изображение.

Выбор метода вставки. В текстовом процессоре Microsoft Word 2010 избранный рисунок можно вставить в документ тремя способами: *внедрением* (**Вставить**), *связыванием* (**Связать с файлом**) и *внедрением со связыванием* (**Вставить и связать**).

1. В первом случае объект войдет в документ и может передаваться вместе с ним.
2. Во втором случае он останется по месту своего хранения, а в документ войдет только указатель на первоисточник.
3. В третьем случае объект войдет в документ, но его связь с первоисточником сохранится. Это полезно, если предполагается возможность редактирования первоисточника и надо обеспечить синхронное редактирование и внедренного объекта.

Выбор метода вставки выполняют в диалоговом окне **Добавление рисунка**. В его правом нижнем углу есть раскрывающийся список, в котором следует выбрать один метод из трех возможных.

При изменении связанного файла изменения отразятся в документе при открытии документа.

Вставленный в документ рисунок можно заменить, выбрав пункт контекстного меню рисунка **Изменить рисунок**.

Взаимодействие изображения с текстом. Основная часть инструментов для настройки свойств изображений в текстовом документе сосредоточена на вкладке **Формат**. При выборе рисунка в тексте документа эта вкладка открывается автоматически.

По способу взаимодействия с текстом выделяют два основных типа изображений: *внедренные в строку* (*inline*) и *свободные* (*floating*). Изображения первого типа можно условно рассматривать как отдельные символы: при движении текста в процессе редактирования изображение перемещается вместе с ним и остается в том месте текста, куда его поместили. Положение свободного изображения на странице не связано с позицией ввода. Изображение взаимодействует с текстом посредством обтекания.

Для управления методом взаимодействия изображения с текстом служит кнопка **Формат ► Упорядочить ► Положение**.

Элемент управления **В тексте** обеспечивает внедрение изображения в текстовую строку. Прочие элементы служат для выбора одного из методов обтекания.

Если изображение вставлено в документ как свободное, дополнительные средства настройки обтекания можно получить из меню, которое открывается кнопкой **Формат ► Упорядочить ► Обтекание текстом**. В частности, здесь присутствует пункт **Изменить контур обтекания**, который позволяет создавать интересные варианты обтекания изображения по криволинейному контуру.

Приемы редактирования изображения. В текстовом процессоре Microsoft Word 2010 имеются встроенные средства редактирования изображений. Они имеют относительно малые возможности и, если говорить строго, их не вполне корректно считать средством редактирования изображений. При их использовании оригинал изображения не меняется, а меняется только способ его отображения в документе. Фактически здесь редактируется не изображение, а фильтр, управляющий тем, как оно выглядит в документе.

На вкладке **Формат** имеются группы элементов **Изменение**, **Стили рисунков** и **Размер**. С помощью их элементов управления можно выполнить следующие действия:

- ☐ Настройка контрастности;
- ☐ Настройка яркости;
- ☐ Настройка резкости;
- ☐ Применение к изображению художественных эффектов;
- ☐ Обрезка;
- ☐ Удаление фона;
- ☐ Установка прозрачного цвета;
- ☐ Настройка цветности.

Функция установки прозрачного цвета имеет особое значение для создания веб-страниц. Она позволяет назначить один (любой) из цветов изображения в качестве «прозрачного». При размещении такого графического объекта поверх других объектов (это выполняется настройкой метода обтекания) все объекты нижележащего слоя видны через те участки верхнего изображения, которые имеют цвет, назначенный прозрачным. Разумеется, изображения, используемые для такого представления, надо готовить особо. Они должны иметь большие участки, окрашенные однородным фоновым цветом. Для этого изображение либо предварительно обрабатывают в графическом редакторе, либо сразу снимают цифровой фотокамерой на однородном фоне (как правило, синего цвета).

Если требуется провести более серьезное редактирование рисунка, следует воспользоваться специализированными программами. Например, в комплект Microsoft Office входит *Диспетчер рисунков Microsoft Office*. Он поддерживает некоторые средства для редактирования графических файлов. Кроме того, можно воспользоваться программами сторонних поставщиков. Для редактирования внедренного изображения средствами внешних программ можно сохранить это изображение в виде внешнего файла (команда **Сохранить как рисунок контекстного меню рисунка**) и, отредактировав внешним приложением, снова внедрить в документ. Рисунки, вставленные в документ методом связи, можно редактировать непосредственно. Изменения таких рисунков будут автоматически отображены в документе.

Практическое занятие

Упражнение 11.1. Создание сложных таблиц методом рисования

На рис. 11.15 представлен фрагмент технологической карты механической обработки детали. По своей сути технологическая карта является табличной формой сложной структуры. В данном упражнении мы рассмотрим процесс ее создания средствами текстового процессора Microsoft Word.

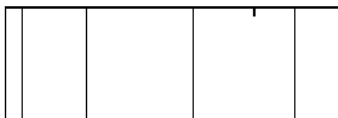
Переход	Содержание перехода	Инструмент (код и наименование)			Режим обработки					T _о	T _в
		вспомогательный	режущий	измерительный	T	i	S	n	V		
A											
1											
2											
3											

Рис. 11.20. Фрагмент карты механической обработки детали

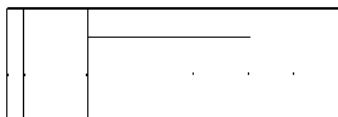
1. Запустите текстовый процессор.
2. Создайте новый документ на базе обычного шаблона.
3. В качестве режима представления документа включите Режим разметки (Вид ► Режимы просмотра документа ► Разметка страницы), чтобы четко видеть границы полосы набора.
4. Откройте вкладку Вставка.
5. Выберите инструмент Таблицы ► Таблица ► Нарисовать таблицу.
6. Методом протягивания нарисуйте с его помощью прямоугольник, ширина которого равна ширине полосы набора. Высота прямоугольника может быть произвольной — его можно будет растянуть или сжать впоследствии. Для этого достаточно навести указатель мыши на нижнюю границу рамки и, когда указатель сменит форму, переместить рамку методом перетаскивания.

Полученный прямоугольник представляет собой внешнюю границу таблицы. Для прочих границ она будет *опорной*, то есть они должны начинаться и заканчиваться на опорной границе.

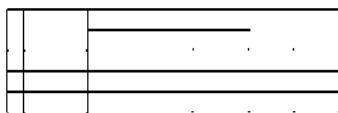
7. Проведите пять вертикальных линий. Это внутренние границы. Они опираются на внешние границы. Для горизонтальных границ, которые будут на них опираться, они будут выполнять функции опорных. На ширину столбцов не обращайте внимания — ее можно будет изменить впоследствии. Сейчас мы разрабатываем только структуру таблицы.



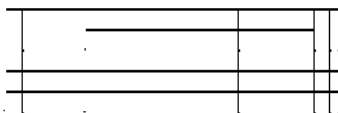
8. Убедитесь, что с помощью инструмента Ластик можно удалить любую из только что проведенных границ. Удаление выполняется одним щелчком. Внешние границы удалить нельзя.
9. Проведите две горизонтальные линии, как показано на рисунке.



10. Убедитесь с помощью Ластика в том, что вертикальные линии, ставшие опорными для первой горизонтальной линии, не могут быть удалены.
11. Выделите всю таблицу. Для этого введите в нее указатель мыши, вызовите контекстное меню и дайте команду **Выделить ► Таблица**.
12. Когда таблица выделена, можно задать высоту ее строк. Для этого нужно вызвать контекстное меню таблицы и выбрать в нем пункт **Свойства таблицы**. В окне следует выбрать вкладку **Строка**. На ней следует воспользоваться группой параметров **Размер** и задать высоту строк.
13. Добавьте в нижней части таблицы несколько строк командой контекстного меню **Вставить ► Вставить строки снизу**. При необходимости впоследствии можно добавить столько строк, сколько надо.



14. Методом перетаскивания вертикальных границ создайте нужное соотношение между шириной столбцов.



15. Проведите дополнительные вертикальные линии инструментом Нарисовать таблицу.

16. Выделите группы столбцов, которые должны иметь равную ширину. Для этого установите указатель мыши над верхней рамкой таблицы и в тот момент, когда он примет форму стрелки, направленной вниз, щелкните левой кнопкой.
17. Выделенные столбцы станут равными по ширине, если щелкнуть на кнопке Макет ► Размер ячейки ► Выровнять ширину столбцов.

18. Если необходимо выровнять высоту строк, их следует выделить и использовать кнопку Макет ► Размер ячейки ► Выровнять высоту строк.
19. Заполните заголовки столбцов таблицы. Гарнитуру шрифта, его размер и начертание задайте с помощью инструментов вкладки Главная.
20. Обратите внимание на то, что в ячейках таблицы имеет значение не только горизонтальное выравнивание, но и вертикальное, поэтому для задания выравнивания заголовков средств вкладки Главная недостаточно.
21. Нужный метод выравнивания (один из девяти) выбирают в палитре, которая доступна на вкладке Макет ► Выравнивание.
22. При вводе заголовка первого столбца в образце использовано вертикальное расположение текста. Это типичный прием для оформления заголовков узких столбцов. Изменение направления текста выполняют с помощью кнопки Макет ► Выравнивание ► Направление текста.
23. Завершив создание таблицы, сохраните документ Word в папке \Документы.

Итог. Мы научились создавать таблицы сложной структуры методом «рисования» и использовать автоматические средства управления шириной столбцов, высотой ячеек и их выравниванием.

Упражнение 11.2. Создание диаграмм на основе таблиц

Ниже представлена таблица с итогами испытания на износ образцов легированных сталей при трении скольжения под нагрузкой в условиях недостаточной смазки. Замеры величины износа образца производились восемь раз через каждые пятнадцать минут.

Пара трения	Износ верхнего образца, мг							
	15 мин	30 мин	45 мин	60 мин	75 мин	90 мин	105 мин	120 мин
40X13/95X18	11,2	7,6	4,2	1,8	1,1	1,2	1,1	1,2
40X13/40XH	17,4	12,5	9,5	7,4	5,3	4,8	4,5	4,4
40XH/95X18	12,1	6,4	3,1	2,2	1,7	1,6	1,6	1,6

В этом упражнении мы построим диаграмму на базе данной таблицы.

1. Запустите текстовый процессор.
2. Создайте новый документ на базе стандартного шаблона.
3. В качестве режима представления документа включите **Режим разметки** (**Вид** ► **Режимы просмотра документа** ► **Разметка страницы**), чтобы четко видеть границы полосы набора.
4. Командой **Вставка** ► **Таблицы** ► **Таблица** создайте базовую таблицу, имеющую 5 строк и 9 столбцов.
5. Выделите две верхние ячейки первого столбца и объедините их командой **Макет** ► **Объединение** ► **Объединить ячейки**.
6. Выделите ячейки первой строки для столбцов со второго по девятый и объедините их.
7. Заполните таблицу согласно прилагаемому образцу.
8. Установите указатель мыши в поле таблицы, дождитесь появления над левым верхним углом таблицы маркера выделения и, щелкнув по нему мышью, выделите таблицу. Скопируйте выделенную таблицу в буфер обмена командой контекстного меню таблицы **Копировать**.
9. Вставьте базовую диаграмму командой **Вставка** ► **Иллюстрация** ► **Диаграмма**. В окне выбора типа диаграммы выберите **График**.
10. Выделите содержимое базовой таблицы диаграммы в Microsoft Excel щелчком на ячейке, образованной на пересечении заголовков строк и столбцов в левом верхнем углу.
11. Замените содержимое базовой таблицы содержимым своей таблицы командой вставки содержимого из буфера обмена (**Главная** ► **Буфер обмена** ► **Вставить**).
12. Обратите внимание на то, как изменилась диаграмма: она пришла в соответствие с содержимым таблицы.
13. Щелкните правой кнопкой мыши на диаграмме и в контекстном меню выберите пункт **Изменить тип диаграммы**. Средствами открывшегося диалогового окна проверьте, как выглядят диаграммы других типов.
14. Закройте диалоговое окно **Тип диаграммы**. Сохраните документ Word в папке \Документы.

Итог. В этом упражнении мы освоили один из двух основных методов создания диаграмм — метод, основанный на использовании базовой таблицы, которая содержится в документе.

Упражнение 11.3. Изучение эффективных приемов работы с графическими объектами

1. Запустите текстовый процессор.
2. Создайте новый документ на базе стандартного шаблона.
3. В качестве режима представления документа включите Режим разметки (Вид ► Режимы просмотра документа ► Разметка страницы), чтобы четко видеть границы полосы набора.
4. Введите несколько строк произвольного текста.
5. Командой Вставка ► Иллюстрации ► Рисунок вставьте ниже текста рисунок из произвольного файла, например из файла Хризантема.jpg из библиотеки Изображения.
6. Выделите рисунок щелчком левой кнопки мыши — откроется вкладка Формат группы Работа с рисунками.
7. Дайте команду Формат ► Упорядочить ► Положение ► В тексте. Передвиньте изображение методом перетаскивания, оценивая происходящее взаимодействие с текстом.
8. Дайте команду Формат ► Упорядочить ► Обтекание текстом ► По контуру. Проверьте, как происходит взаимодействие с текстом при перемещении изображения.
9. Выделите изображение, скопируйте его в буфер обмена (CTRL+C) и создайте рядом его копию (CTRL+V).
10. Перемещая оба изображения, добейтесь их положения рядом, с выравниванием по верхнему краю.
11. Повторите перемещение изображений с выравниванием при нажатой клавише ALT. Убедитесь в том, что перемещение изображений происходит дискретно, с привязкой к узлам невидимой сетки, что позволяет выполнить выравнивание абсолютно точно.
12. Выделите одно из изображений. Используя угловой маркер, измените его размер методом перетаскивания.
13. Восстановите прежний размер изображения.
14. Повторите перетаскивание углового маркера, но при нажатой клавише CTRL. Обратите внимание на то, что характер изменения размера изображения изменился. В данном случае оно перемасштабируется «от центра».
15. Сохраните итоговый документ Word в папке \Документы.

Итог. Мы освоили два основных приема вставки изображения в текст — с внедрением в строку и со свободным размещением. Мы убедились, что использование клавиш CTRL и ALT при работе с изображениями в документе открывает дополнительные возможности оформления.

Упражнение 11.4. Создание графических заголовков

Для создания художественных графических надписей, например заголовков, текстовый процессор Microsoft Word 2010 имеет специальное программное средство WordArt. Доступ к нему осуществляется посредством команды Вставка ► Текст ► WordArt.

Графические объекты, вставленные в текстовый документ средством WordArt, могут распечатываться вместе с документом на выводном печатающем устройстве, могут отображаться в составе электронного документа, распространяемого в формате Microsoft Word, и могут отображаться на веб-страницах. Однако при экспорте документа в форматы других программ, предназначенных для обработки документов, объекты WordArt не всегда воспроизводятся правильно, то есть при создании документов, в которых содержание играет более высокую роль, чем оформление, использовать художественные заголовки, выполненные средствами WordArt, не рекомендуется.

1. Запустите текстовый процессор.
2. Создайте новый документ на базе стандартного шаблона.
3. В качестве режима представления документа включите Режим разметки (Вид ► Режимы просмотра документа ► Разметка страницы), чтобы четко видеть границы полосы набора.
4. Введите несколько строк произвольного текста, скопируйте их в буфер обмена.
5. Щелкните на кнопке Вставка ► Текст ► WordArt и в появившемся меню выберите желаемый стиль оформления надписи.
6. В появившемся поле с надписью «Поместите здесь ваш текст» вставьте скопированный ранее текст.
7. Воспользуйтесь вкладкой Средства рисования ► Формат для настройки визуальных параметров вставленного текста.
8. Поэкспериментируйте со средствами оформления текста, помимо вкладки Формат используйте стандартные инструменты форматирования текста с вкладки Главная. Проверьте, как протекают следующие операции (после каждой команды возвращайтесь к исходному состоянию комбинацией CTRL+Z):
 - изменение содержания надписи (щелчок мышью в поле текста);
 - изменение стиля оформления (вкладка Формат);
 - изменение характера взаимодействия с основным текстом (Формат ► Упорядочить ► Положение);
 - изменение формы надписи (Формат ► Стили WordArt ► Анимация ► Преобразовать);
 - выравнивание букв надписи по высоте (Формат ► Текст);
 - расположение текста надписи по вертикали (Формат ► Текст);
 - управление интервалом между символами (Главная ► Шрифт ► Шрифт ► Дополнительно ► Интервал).

9. Закончив эксперименты, создайте заголовок по своему вкусу и сохраните документ Word в папке \Документы.

Итог. Мы научились создавать художественные заголовки, внедрять их в документы и редактировать «по месту». В то же время мы узнали, что для документов, передаваемых на последующую обработку, пользоваться этим средством не рекомендуется.

Глава двенадцатая

Обработка данных средствами электронных таблиц

Для представления данных в удобном виде используют таблицы. Компьютер позволяет представлять их в электронной форме, а это дает возможность не только отображать, но и обрабатывать данные. Класс программ, используемых для этой цели, называется *электронными таблицами*.

Особенность электронных таблиц заключается в возможности применения формул для описания связи между значениями различных ячеек. Расчет по заданным формулам выполняется автоматически. Изменение содержимого какой-либо ячейки приводит к пересчету значений всех ячеек, которые с ней связаны формульными отношениями, и тем самым к обновлению всей таблицы в соответствии с изменившимися данными.

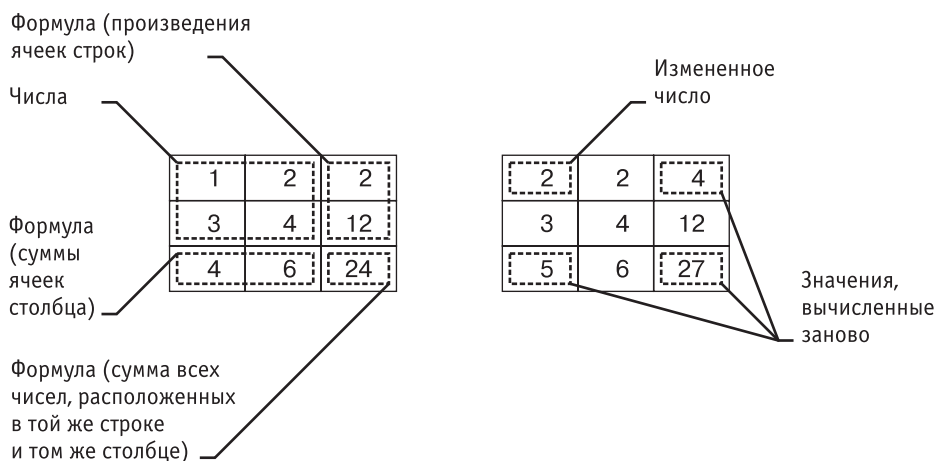


Рис. 12.1. При изменении содержания одной из ячеек таблицы все формулы пересчитываются и значения в ячейках, которые прямо или косвенно зависят от измененных, автоматически обновляются

Применение электронных таблиц упрощает работу с данными и позволяет получать результаты без проведения расчетов вручную или специального программирования. Наиболее широкое применение электронные таблицы нашли в экономических и бухгалтерских расчетах, но и в научно-технических задачах электронные таблицы можно использовать эффективно, например, для:

- ☐ проведения однотипных расчетов над большими наборами данных;
- ☐ автоматизации итоговых вычислений;
- ☐ решения задач путем подбора значений параметров, табулирования формул;
- ☐ обработки результатов экспериментов;

- ❑ проведения поиска оптимальных значений параметров;
- ❑ подготовки табличных документов;
- ❑ построения диаграмм и графиков по имеющимся данным.

Одним из наиболее распространенных средств работы с документами, имеющими табличную структуру, является программа Microsoft Excel.

12.1. Основные понятия электронных таблиц

Программа Microsoft Excel предназначена для работы с таблицами данных, преимущественно числовых. При формировании таблицы выполняют ввод, редактирование и форматирование текстовых и числовых данных, а также формул. Наличие средств автоматизации облегчает эти операции. Созданная таблица может быть выведена на печать.

Рабочая книга и рабочий лист. Строки, столбцы, ячейки

Документ Excel называется *рабочей книгой*. Рабочая книга представляет собой набор *рабочих листов*, каждый из которых имеет табличную структуру и может содержать одну или несколько таблиц. В окне документа в программе Excel отображается только *текущий* рабочий лист, с которым и ведется работа (рис. 12.2). Каждый рабочий лист имеет *название*, которое отображается на ярлычке *листа*, отображаемом в его нижней части. С помощью ярлычков можно переключаться к другим рабочим листам, входящим в ту же самую рабочую книгу. Чтобы переименовать рабочий лист, надо дважды щелкнуть на его ярлычке.

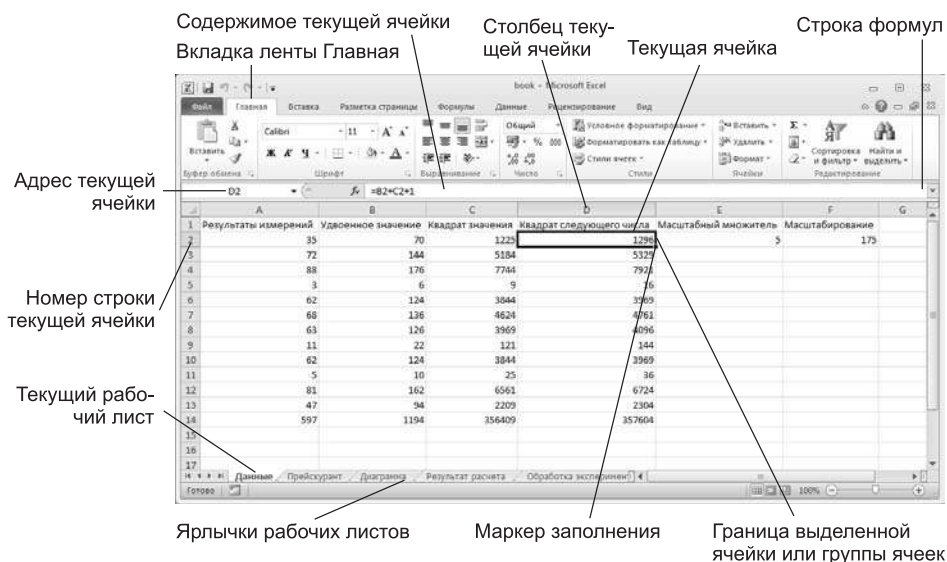


Рис. 12.2. Рабочий лист электронной таблицы Excel

Рабочий лист состоит из *строк* и *столбцов*. Столбцы озаглавлены прописными латинскими буквами и, далее, двухбуквенными комбинациями. В ранних версиях программы, до Microsoft Excel 2007, рабочий лист мог содержать до 256 столбцов и до 65 536 строк. В современных версиях программы действуют ограничения на 1 048 576 строк и 16 384 столбцов. Нумерация столбцов начинается с А, строки нумеруются с 1.

Ячейки и их адресация. На пересечении столбцов и строк образуются *ячейки* таблицы. Они являются минимальными элементами для хранения данных. Обозначение отдельной ячейки сочетает в себе номера столбца и строки (в этом порядке), на пересечении которых она расположена, например: А1 или DE234. Обозначение ячейки (ее номер) выполняет функции ее адреса. Адреса ячеек используются при записи формул, определяющих взаимосвязь между значениями, расположенными в разных ячейках.

Одна из ячеек всегда является *активной* и выделяется *рамкой активной ячейки*. Эта рамка в программе Excel играет роль курсора. Операции ввода и редактирования всегда производятся в активной ячейке. Переместить рамку активной ячейки можно с помощью курсорных клавиш или указателя мыши.

Диапазон ячеек. На данные, расположенные в соседних ячейках, можно ссылаться в формулах как на единое целое. Такую группу ячеек называют *диапазоном*. Наиболее часто используют прямоугольные диапазоны, образующиеся на пересечении группы последовательно идущих строк и группы последовательно идущих столбцов. Диапазон ячеек обозначают, указывая через двоеточие номера ячеек, расположенных в противоположных углах прямоугольника, например: А1:С15.

Если требуется выделить прямоугольный диапазон ячеек, это можно сделать протягиванием указателя от одной угловой ячейки до противоположной по диагонали. Рамка текущей ячейки при этом расширяется, охватывая весь выбранный диапазон. Чтобы выбрать столбец или строку целиком, следует щелкнуть на заголовке столбца (строки). Протягиванием указателя по заголовкам можно выбрать несколько идущих подряд столбцов или строк.

Ввод, редактирование и форматирование данных

Отдельная ячейка может содержать данные, относящиеся к одному из трех типов: *текст*, *число* или *формула*, а также оставаться пустой. Программа Excel при сохранении рабочей книги записывает в файл только прямоугольную область рабочих листов, примыкающую к левому верхнему углу (ячейка А1) и содержащую все заполненные ячейки.

Тип данных, размещаемых в ячейке, определяется автоматически при вводе. Если эти данные можно интерпретировать как число, программа Excel так и делает. В противном случае данные рассматриваются как текст. Ввод формулы всегда начинается с символа «=» (знака равенства).

Ввод текста и чисел. Ввод данных осуществляют непосредственно в текущую ячейку или в *строку формул*, располагающуюся в верхней части окна программы под лентой (см. рис. 12.2). Место ввода отмечается текстовым курсором. Если начать ввод нажатием алфавитно-цифровых клавиш, данные из текущей ячейки

заменяются вводимым текстом. Если щелкнуть на строке формул или дважды на текущей ячейке, старое содержимое ячейки не удаляется и появляется возможность его редактирования. Вводимые данные в любом случае отображаются как в ячейке, так и в строке формул.

Чтобы завершить ввод, сохранив введенные данные, используют кнопку **Ввод** в строке формул или клавишу **ENTER**. Чтобы отменить внесенные изменения и восстановить прежнее значение ячейки, используют кнопку **Отмена** в строке формул или клавишу **ESC**. Для очистки текущей ячейки или выделенного диапазона проще всего использовать клавишу **DELETE**.

Форматирование содержимого ячеек. Текстовые данные по умолчанию выравниваются по левому краю ячейки, а числа — по правому. Чтобы изменить формат отображения данных в текущей ячейке или выбранном диапазоне, используют команды вкладки ленты **Главная**, а также окно **Формат ячеек**, которое можно вызвать из контекстного меню выделенного диапазона ячеек командой **Формат ячеек**. Вкладки этого диалогового окна позволяют выбирать формат записи данных (количество знаков после запятой, указание денежной единицы, способ записи даты и пр.), задавать направление текста и метод его выравнивания, определять шрифт и начертание символов, управлять отображением и видом рамок, задавать фоновый цвет.

12.2. Содержание электронной таблицы

Формулы

Вычисления в таблицах программы Excel осуществляются при помощи *формул*. Формула может содержать числовые константы, ссылки на ячейки и *функции* Excel, соединенные знаками математических операций. Скобки позволяют изменять стандартный порядок выполнения действий. Если ячейка содержит формулу, то в рабочем листе отображается текущий результат вычисления этой формулы. Если сделать ячейку текущей, то сама формула отображается в строке формул.

Правило использования формул в программе Excel состоит в том, что, если значение ячейки *действительно* зависит от других ячеек таблицы, *всегда* следует использовать формулу, даже если операцию легко можно выполнить в «уме». Это гарантирует, что последующее редактирование таблицы не нарушит ее целостности и правильности производимых в ней вычислений.

Ссылки на ячейки

Формула может содержать *ссылки*, то есть адреса ячеек, содержимое которых используется в вычислениях. Это означает, что результат вычисления формулы зависит от числа, находящегося в другой ячейке. Ячейка, содержащая формулу, таким образом, является *зависимой*. Значение, отображаемое в ячейке с формулой, пересчитывается при изменении значения ячейки, на которую указывает ссылка.

Ссылку на ячейку можно задать разными способами. Во-первых, адрес ячейки можно ввести вручную. Другой способ состоит в щелчке на нужной ячейке или

выборе диапазона, адрес которого требуется ввести. Ячейка или диапазон при этом выделяются пунктирной рамкой.

Все диалоговые окна программы Excel, которые требуют указания номеров или диапазонов ячеек, содержат кнопки, присоединенные к соответствующим полям. При щелчке на такой кнопке диалоговое окно сворачивается до минимально возможного размера, что облегчает выбор нужной ячейки (диапазона) с помощью щелчка или протягивания (рис. 12.3).

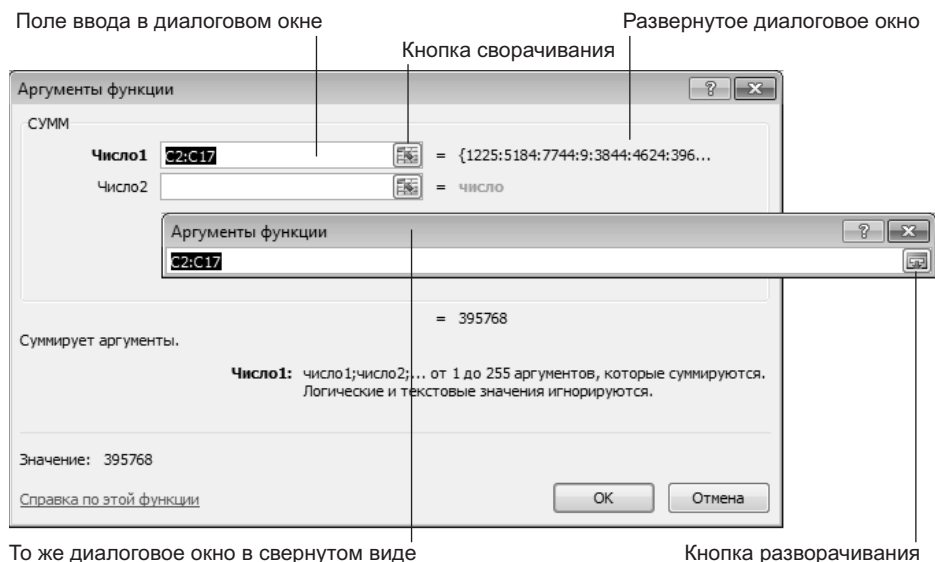


Рис. 12.3. Диалоговое окно в развернутом и свернутом виде

Для редактирования формулы следует дважды щелкнуть на соответствующей ячейке. При этом ячейки (диапазоны), от которых зависит значение формулы, выделяются на рабочем листе цветными рамками, а сами ссылки отображаются в ячейке и в строке формул тем же цветом. Это облегчает редактирование и проверку правильности формул.

Абсолютные и относительные ссылки

По умолчанию ссылки на ячейки в формулах рассматриваются как *относительные*. Это означает, что при копировании формулы адреса в ссылках автоматически изменяются в соответствии с относительным расположением исходной ячейки и создаваемой копии.

Пусть, например, в ячейке B2 имеется ссылка на ячейку A3. В относительном представлении можно сказать, что ссылка указывает на ячейку, которая располагается на один столбец левее и на одну строку ниже данной. Если формула будет скопирована в другую ячейку, то такое относительное указание ссылки сохранится. Например, при копировании формулы в ячейку EA27 ссылка будет продолжать

указывать на ячейку, располагающуюся левее и ниже, в данном случае на ячейку DZ28.

При *абсолютной адресации* адреса ссылок при копировании не изменяются, так что ячейка, на которую указывает ссылка, рассматривается как нетабличная. Для изменения способа адресации при редактировании формулы надо выделить ссылку на ячейку и нажать клавишу F4. Элементы номера ячейки, использующие абсолютную адресацию, предваряются символом \$. Например, при последовательных нажатиях клавиши F4 номер ячейки A1 будет записываться как A1, \$A\$1, A\$1 и \$A1. В двух последних случаях один из компонентов номера ячейки рассматривается как абсолютный, а другой — как относительный.

Копирование содержимого ячеек

Копирование и перемещение ячеек в программе Excel можно осуществлять методом перетаскивания или через буфер обмена. При работе с небольшим числом ячеек удобно использовать первый метод, при работе с большими диапазонами — второй.

Метод перетаскивания. Чтобы методом перетаскивания скопировать или переместить текущую ячейку (выделенный диапазон) вместе с содержимым, следует навести указатель мыши на рамку текущей ячейки (он примет вид стрелки с дополнительными стрелочками). Теперь ячейку можно перетащить в любое место рабочего листа (точка вставки помечается всплывающей подсказкой).

Для выбора способа выполнения этой операции, а также для более надежного контроля над ней рекомендуется использовать *специальное перетаскивание* с помощью правой кнопки мыши. В этом случае при отпускании кнопки мыши появляется специальное меню, в котором можно выбрать конкретную выполняемую операцию.

Применение буфера обмена. Передача информации через буфер обмена имеет в программе Excel определенные особенности, связанные со сложностью контроля над этой операцией. Вначале необходимо выделить копируемый (вырезаемый) диапазон и дать команду на его помещение в буфер обмена: Главная ► Буфер обмена ► Копировать или Главная ► Буфер обмена ► Вырезать. Вставка данных в рабочий лист возможна лишь немедленно после их помещения в буфер обмена. Попытка выполнить любую другую операцию приводит к отмене начатого процесса копирования или перемещения. Однако утраты данных не происходит, поскольку «вырезанные» данные удаляются из места их исходного размещения только в момент выполнения вставки.

Место вставки определяется путем указания ячейки, соответствующей верхнему левому углу диапазона, помещенного в буфер обмена, или путем выделения диапазона, который по размерам в точности равен копируемому (перемещаемому). Вставка выполняется командой Главная ► Буфер обмена ► Вставить. Для управления способом вставки можно использовать команду Главная ► Буфер обмена ► Вставить ► Специальная вставка. В этом случае правила вставки данных из буфера обмена задаются в открывшемся диалоговом окне. Для копирования, вырезания, вставки и специальной вставки можно пользоваться, помимо вышеописанных команд, командами контекстного меню ячеек.

Автоматизация ввода

Так как таблицы часто содержат повторяющиеся или однотипные данные, программа Excel содержит средства автоматизации ввода. К числу предоставляемых средств относятся: *автозавершение*, *автозаполнение числами* и *автозаполнение формулами*.

Автозавершение. Для автоматизации ввода текстовых данных используется метод *автозавершения*. Его применяют при вводе в ячейки одного столбца рабочего листа текстовых строк, среди которых есть повторяющиеся. В ходе ввода текстовых данных в очередную ячейку программа Excel проверяет соответствие введенных символов строкам, имеющимся в этом столбце выше. Если обнаружено однозначное совпадение, введенный текст автоматически дополняется. Нажатие клавиши ENTER подтверждает операцию автозавершения, в противном случае ввод можно продолжать, не обращая внимания на предлагаемый вариант.

Можно прервать работу средства автозавершения, оставив в столбце пустую ячейку. И наоборот, чтобы использовать возможности средства автозавершения, заполненные ячейки должны идти подряд, без промежутков между ними.

Автозаполнение числами. При работе с числами используется метод *автозаполнения*. В правом нижнем углу рамки текущей ячейки имеется черный квадратик — *маркер заполнения*. При наведении на него указатель мыши (он обычно имеет вид толстого белого креста) приобретает форму тонкого черного крестика. Перетаскивание маркера заполнения рассматривается как операция «размножения» содержимого ячейки в горизонтальном или вертикальном направлении.

Если ячейка содержит число (в том числе дату, денежную сумму), то при перетаскивании маркера происходит копирование ячеек или их заполнение арифметической прогрессией. Для выбора способа автозаполнения следует производить специальное перетаскивание с использованием правой кнопки мыши.

Пусть, например, ячейка A1 содержит число 1. Наведите указатель мыши на маркер заполнения, нажмите правую кнопку мыши и перетащите маркер заполнения так, чтобы рамка охватила ячейки A1, B1 и C1, и отпустите кнопку мыши. Если теперь выбрать в открывшемся меню пункт *Копировать ячейки*, все ячейки будут содержать число 1. Если же выбрать пункт *Заполнить*, то в ячейках окажутся числа 1, 2 и 3.

Чтобы точно сформулировать условия заполнения ячеек, следует дать команду Главная ► Редактирование ► Заполнить ► Прогрессия. В открывшемся диалоговом окне Прогрессия выбирается тип прогрессии, величина шага и предельное значение. После щелчка на кнопке ОК программа Excel автоматически заполняет ячейки в соответствии с заданными правилами.

Автозаполнение формулами. Эта операция выполняется так же, как автозаполнение числами. Ее особенность заключается в необходимости копирования ссылок на другие ячейки. В ходе автозаполнения во внимание принимается характер ссылок в формуле: относительные ссылки изменяются в соответствии с относительным расположением копии и оригинала, абсолютные остаются без изменений.

Для примера предположим, что значения в третьем столбце рабочего листа (столбце C) вычисляются как суммы значений в соответствующих ячейках столб-

цов А и В. Введем в ячейку С1 формулу $=A1+B1$. Теперь скопируем эту формулу методом автозаполнения во все ячейки третьего столбца таблицы. Благодаря относительной адресации формула будет правильной для всех ячеек данного столбца.

В табл. 12.1 приведены правила обновления ссылок при автозаполнении вдоль строки или вдоль столбца.

Таблица 12.1. Правила обновления ссылок при автозаполнении

Ссылка в исходной ячейке	Ссылка в следующей ячейке	
	При заполнении вправо	При заполнении вниз
A1 (относительная)	B1	A2
\$A1 (абсолютная по столбцу)	\$A1	\$A2
A\$1 (абсолютная по строке)	B\$1	A\$1
\$A\$1 (абсолютная)	\$A\$1	\$A\$1

Использование стандартных функций

Стандартные функции используются в программе Excel только в формулах. *Вызов функции* состоит в указании в формуле *имени функции*, после которого в скобках указывается *список параметров*. Отдельные параметры разделяются в списке точкой с запятой. В качестве параметра может использоваться число, адрес ячейки или произвольное выражение, для вычисления которого также могут использоваться функции.

В режиме ввода формулы в левой части строки формул, где раньше располагался номер текущей ячейки, появляется раскрывающийся список функций. Он содержит десять функций, которые использовались последними, а также пункт *Другие функции*.

Использование мастера функций. При выборе пункта *Другие функции* запускается Мастер функций, облегчающий выбор нужной функции. В раскрывающемся списке *Категория* выбирается категория, к которой относится функция (если определить категорию затруднительно, используют пункт *Полный алфавитный перечень*), а в списке *Выберите функцию* — конкретная функция данной категории. После щелчка на кнопке *ОК* имя функции заносится в строку формул вместе со скобками, ограничивающими список параметров. Текстовый курсор устанавливается между этими скобками. Вызвать Мастер функций можно и проще, щелчком на кнопке *Вставка функции* в строке формул.

Аргументы функции. Как только имя функции выбрано, на экране появляется диалоговое окно *Аргументы функции*. Это окно, в частности, содержит значение, которое получится, если немедленно закончить ввод формулы (рис. 12.4).

Правила вычисления формул, содержащих функции, не отличаются от правил вычисления более простых формул. Ссылки на ячейки, используемые в качестве параметров функции, также могут быть относительными или абсолютными, что учитывается при копировании формул методом автозаполнения.



Рис. 12.4. Строка формул и диалоговое окно Аргументы функции

12.3. Печать документов Excel

Экранное представление электронной таблицы в Excel значительно отличается от того, которое получилось бы при выводе данных на печать. Это связано с тем, что единый рабочий лист приходится разбивать на фрагменты, размер которых определяется форматом печатного листа. Кроме того, элементы оформления рабочего окна программы: номера строк и столбцов, условные границы ячеек — обычно не отображаются при печати.

Для того чтобы приступить к печати рабочего листа, следует выполнить команду **Файл ► Печать**. В открытом окне **Печать**, в его правой части, находится область предварительного просмотра документа (рис. 12.5).

Окно предварительного просмотра не допускает редактирования документа, но позволяет увидеть его на экране точно в таком виде, в каком он будет напечатан.

При необходимости до распечатки документа можно изменить свойства страницы и параметры печати. Настройки выполняются в левой части окна **Печать**.

Для настройки параметров можно, во-первых, воспользоваться либо элементами управления окна **Печать**, либо вызвав окно **Параметры страницы** с помощью соответствующей ссылки.

Списки настроек левой части окна **Печать** позволяют задавать область печати, ориентацию, формат страницы, настраивать поля и масштаб страницы. Использование этих элементов управления позволяет достаточно быстро настроить основные параметры печати.

Окно **Параметры страницы** дает доступ к детальным настройкам. В частности, это окно позволяет задать следующие параметры страницы: ориентацию листа,

масштаб страницы (изменение масштаба позволяет управлять числом печатных страниц, необходимых для документа), размеры полей документа. Здесь же можно задать верхние и нижние колонтитулы для страницы. На вкладке Лист включается или отключается печать сетки и номеров строк и столбцов, а также выбирается последовательность разбиения на страницы рабочего листа, превосходящего размеры печатной страницы как по длине, так и по ширине.

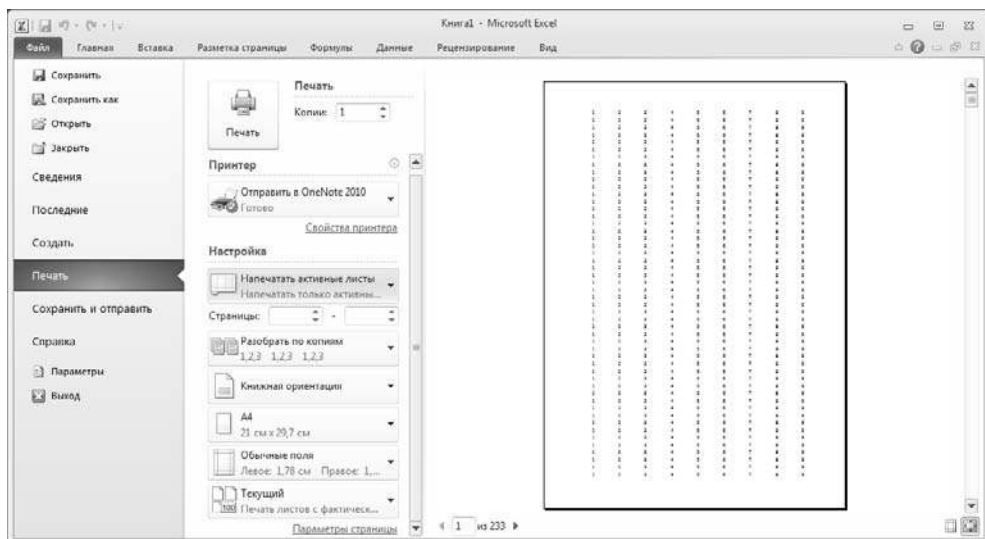


Рис. 12.5. Предварительный просмотр документа перед печатью и настройки печати

Настроить параметры печатной страницы можно в процессе работы с документом, используя инструменты вкладки ленты **Разметка страницы**. Документы, подготовленные в Microsoft Excel, чаще всего используют в электронном виде. Однако, если документ изначально предназначен для печати, можно работать с рабочими листами в режиме разметки страницы (**Вид** ► **Режимы просмотра книги** ► **Разметка страницы**). Этот режим позволяет контролировать вид будущих печатных страниц все время работы с книгой.

После выполнения необходимых настроек можно перейти к печати документа. Для этого достаточно нажать на кнопку **Печать**, которая расположена в верхнем левом углу окна **Печать**.

Выбор области печати

Область печати — это часть рабочего листа, которая должна быть выведена на печать. По умолчанию область печати совпадает с заполненной частью рабочего листа и представляет собой прямоугольник, примыкающий к верхнему левому углу рабочего листа и захватывающий все заполненные ячейки. Если часть данных не должна выводиться на бумагу, область печати можно задать вручную. Для этого надо выделить ячейки, которые должны быть включены в область печати, и дать команду **Разметка страницы** ► **Параметры страницы** ► **Область печати** ► **Задать**. Если

текущей является одна-единственная ячейка, то программа предполагает, что область печати не выделена, и выдает предупреждающее сообщение.

Если область печати задана, то программа отображает в режиме предварительного просмотра и распечатывает только ее. Границы области печати выделяются на рабочем листе крупным пунктиром (сплошной линией в режиме разметки). Для изменения области печати можно задать новую область или при помощи команды Разметка страницы ► Параметры страницы ► Область печати ► Убрать вернуться к параметрам, используемым по умолчанию.

Границы отдельных печатных страниц отображаются на рабочем листе мелким пунктиром. В некоторых случаях требуется, чтобы определенные ячейки располагались вместе на одной и той же печатной странице или, наоборот, разделение печатных страниц происходило в определенном месте рабочего листа. Такая возможность реализуется путем задания границ печатных страниц вручную. Чтобы вставить разрыв страницы, надо сделать текущей ячейку, которая будет располагаться в левом верхнем углу печатной страницы, и дать команду Разметка страницы ► Параметры страницы ► Разрывы ► Вставить разрыв страницы. Программа Excel вставит принудительные разрывы страницы перед строкой и столбцом, в которых располагается данная ячейка. Если выбранная ячейка находится в первой строке или столбце А, то разрыв страницы задается только по одному направлению.

12.4. Применение электронных таблиц для расчетов

В научно-технической деятельности программу Excel трудно рассматривать как основной вычислительный инструмент. Однако ее удобно применять в тех случаях, когда требуется быстрая обработка больших объемов данных. Она полезна для выполнения таких операций, как статистическая обработка и анализ данных, решение задач оптимизации, построение диаграмм и графиков. Для такого рода задач применяют как основные средства программы Excel, так и дополнительные (надстройки).

Итоговые вычисления

Итоговые вычисления предполагают получение числовых характеристик, описывающих определенный набор данных в целом. Например, возможно вычисление суммы значений, входящих в набор, среднего значения и других статистических характеристик, количества или доли элементов набора, удовлетворяющих определенным условиям. Проведение итоговых вычислений в программе Excel выполняется при помощи встроенных функций. Особенность использования таких *итоговых функций* состоит в том, что при их задании программа пытается «угадать», в каких ячейках заключен обрабатываемый набор данных, и задать параметры функции автоматически.

В качестве параметра итоговой функции обычно задается некоторый диапазон ячеек, размер которого определяется автоматически. Выбранный диапазон рассматривается как отдельный параметр («массив»), и в вычислениях используются все ячейки, составляющие его.

Суммирование. Для итоговых вычислений применяют ограниченный набор функций, наиболее типичной из которых является функция суммирования (СУММ). Доступ к этой и к некоторым другим функциям можно получить непосредственно с вкладки Главная (Главная ► Редактирование).

Диапазон суммирования, выбираемый автоматически, включает ячейки с данными, расположенные над текущей ячейкой (предпочтительнее) или слева от нее и образующие непрерывный блок. При неоднозначности выбора используется диапазон, непосредственно примыкающий к текущей ячейке.

Автоматический подбор диапазона не исключает возможности редактирования формулы. Можно переопределить диапазон, который был выбран автоматически, а также задать дополнительные параметры функции.

Функции для итоговых вычислений. Прочие функции для итоговых вычислений могут быть либо выбраны из того же меню, что и функция суммирования, либо введены обычным образом.

Все эти функции относятся к категории Статистические. В их число входят функции ДИСП (вычисляет дисперсию), МАКС (максимальное число в диапазоне), СРЗНАЧ (среднее арифметическое значение чисел диапазона), СЧЕТ (подсчет ячеек с числами в диапазоне) и др.

Функции, предназначенные для выполнения итоговых вычислений, часто применяют при использовании таблицы Excel в качестве базы данных, а именно на фоне фильтрации записей или при создании сводных таблиц.

Использование надстроек

Надстройки — это специальные средства, расширяющие возможности программы Excel. На практике именно надстройки делают программу Excel удобной для использования в научно-технической работе. Хотя эти средства считаются внешними, дополнительными, доступ к ним осуществляется при помощи обычных команд вкладок ленты (обычно через вкладки **Формулы** или **Данные**). Команда использования надстройки обычно открывает специальное диалоговое окно, оформление которого не отличается от стандартных диалоговых окон программы Excel.

Подключить или отключить установленные надстройки можно с помощью команд диалогового окна **Файл ► Параметры ► Надстройки** (рис. 12.6). Подключение надстроек увеличивает нагрузку на вычислительную систему, поэтому обычно рекомендуют подключать только те надстройки, которые реально используются.

Вот основные надстройки, поставляемые вместе с программой Excel.

Пакет анализа. Обеспечивает дополнительные возможности анализа наборов данных. Выбор конкретного метода анализа осуществляется в диалоговом окне **Анализ данных**, которое открывается командой **Данные ► Анализ ► Анализ данных**).

Поиск решения. Эта надстройка используется для решения задач оптимизации. Ячейки, для которых подбираются оптимальные значения и задаются ограничения, выбираются в диалоговом окне **Параметры поиска решения**, которое открывают при помощи команды **Данные ► Анализ ► Поиск решения**.

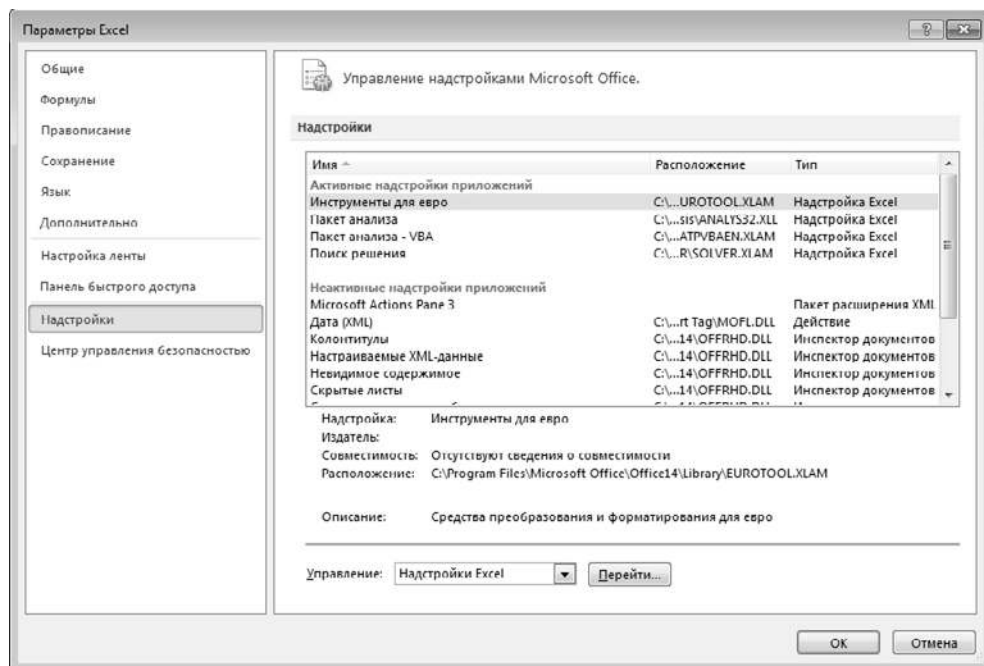


Рис. 12.6. Диалоговое окно для подключения и отключения надстроек

12.5. Построение диаграмм и графиков

В программе Excel термин «диаграмма» используется для обозначения всех видов графического представления числовых данных. Построение графического изображения производится на основе *ряда данных*. Так называют группу ячеек с данными в пределах отдельной строки или столбца. На одной диаграмме можно отображать несколько рядов данных.

Диаграмма представляет собой вставной объект, внедренный на один из листов рабочей книги. Она может располагаться на том же листе, на котором находятся данные, или на любом другом листе (часто для отображения диаграммы отводят отдельный лист). Диаграмма сохраняет связь с данными, на основе которых она построена, и при обновлении этих данных немедленно изменяет свой вид.

Для построения диаграммы обычно используют окно **Вставка диаграммы**, запускаемое командой **Вставка ► Диаграммы ► Создать диаграмму**. Элементы управления для создания отдельных видов диаграмм доступны в группе **Вставка ► Диаграммы**.

Часто удобно заранее выделить область, содержащую данные, которые будут отображаться на диаграмме, но задать эту информацию можно и после вставки диаграммы.

Выбор типа диаграммы

При использовании окна Вставка диаграммы выбор ее типа осуществляется именно в этом окне. При использовании кнопок быстрой вставки диаграмм выбор осуществляется при вставке, без вывода отдельного окна.

Выбор данных

Если диапазон данных был выбран заранее, то в книгу будет добавлена диаграмма, визуализирующая выбранные данные. Если же данные выбраны не были, их можно задать командой контекстного меню диаграммы **Выбрать данные**.

Кроме того, задать набор данных для диаграммы можно командой группы Работа с диаграммами Конструктор ► Данные ► Выбрать данные. При выборе данных для диаграммы появляется окно Выбор источника данных (рис. 12.7).

Если данные не образуют единой группы, то информацию для отрисовки отдельных рядов данных задают, добавляя новые ряды в поле этого окна Элементы легенды (ряды).

Вставленную диаграмму можно изменить. Она состоит из набора отдельных элементов, таких как сами графики (ряды данных), оси координат, заголовков диаграммы, область построения и пр. При щелчке на элементе диаграммы он выделяется маркерами, а при наведении на него указателя мыши — описывается всплывающей подсказкой.

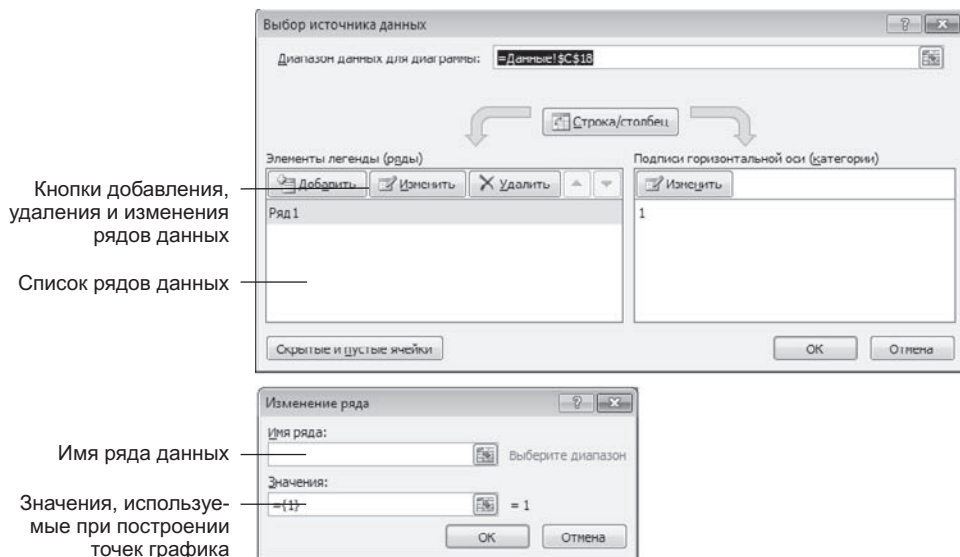


Рис. 12.7. Выбор данных, отображаемых на диаграмме

Практическое занятие

Упражнение 12.1. Обработка данных

1. Запустите программу Excel (Пуск ► Все программы ► Microsoft Office ► Microsoft Excel 2010).
2. Дважды щелкните на ярлычке текущего рабочего листа и дайте этому рабочему листу имя **Данные**.
Дайте команду **Файл ► Сохранить** и сохраните рабочую книгу под именем **book.xlsx** (формат Книга Excel).
3. Сделайте текущей ячейку **A1** и введите в нее заголовок **Результаты измерений**.
4. Введите произвольные числа в последовательные ячейки столбца **A**, начиная с ячейки **A2**.
5. Введите в ячейку **B1** строку **Удвоенное значение**.
6. Введите в ячейку **C1** строку **Квадрат значения**.
7. Введите в ячейку **D1** строку **Квадрат следующего числа**.
8. Введите в ячейку **B2** формулу $=2*A2$.
9. Введите в ячейку **C2** формулу $=A2*A2$.
10. Введите в ячейку **D2** формулу $=B2+C2+1$.
11. Выделите протягиванием ячейки **B2**, **C2** и **D2**.
12. Наведите указатель мыши на маркер заполнения в правом нижнем углу рамки, охватывающей выделенный диапазон. Нажмите левую кнопку мыши и перетаскивайте этот маркер, чтобы рамка охватила столько строк в столбцах **B**, **C** и **D**, сколько имеется чисел в столбце **A**.
13. Убедитесь, что формулы автоматически модифицируются так, чтобы работать со значением ячейки в столбце **A** текущей строки.
14. Измените одно из значений в столбце **A** и убедитесь, что соответствующие значения в столбцах **B**, **C** и **D** в этой же строке были автоматически пересчитаны.
15. Введите в ячейку **E1** строку **Масштабный множитель**.
16. Введите в ячейку **E2** число 5.
17. Введите в ячейку **F1** строку **Масштабирование**.
18. Введите в ячейку **F2** формулу $=A2*E2$.
19. Используйте метод автозаполнения, чтобы скопировать эту формулу в ячейки столбца **F**, соответствующие заполненным ячейкам столбца **A**.
20. Убедитесь, что результат масштабирования оказался неверным. Это связано с тем, что адрес **E2** в формуле задан относительной ссылкой.
21. Щелкните на ячейке **F2**, затем в строке формул. Установите текстовый курсор на ссылку **E2** и нажмите клавишу **F4**. Убедитесь, что формула теперь выглядит как $=A2*\$E\$2$, и нажмите клавишу **ENTER**.
22. Повторите заполнение столбца **F** формулой из ячейки **F2**.

23. Убедитесь, что благодаря использованию абсолютной адресации значения ячеек столбца F теперь вычисляются правильно. Сохраните рабочую книгу book.xlsx.

Итог. Мы научились вводить текстовые и числовые данные в электронные таблицы Excel. Мы узнали, как производится ввод и вычисление формул. Мы также выяснили, как осуществляется копирование формул методом автозаполнения, и определили, в каких случаях следует использовать относительные и абсолютные ссылки.

Упражнение 12.2. Применение итоговых функций

1. Запустите программу Excel (Пуск ► Все программы ► Microsoft Office ► Microsoft Excel 2010) и откройте рабочую книгу book.xlsx, созданную ранее.
2. Выберите рабочий лист Данные.
3. Сделайте текущей первую свободную ячейку в столбце A.
4. Щелкните на кнопке Главная ► Редактирование ► Сумма.
5. Убедитесь, что программа автоматически подставила в формулу функцию СУММ и правильно выбрала диапазон ячеек для суммирования. Нажмите клавишу ENTER.
6. Сделайте текущей следующую свободную ячейку в столбце A.
7. Щелкните на кнопке Вставить функцию в строке формул.
8. В раскрывающемся списке Категория выберите пункт Статистические.
9. В списке Функция выберите функцию СРЗНАЧ и щелкните на кнопке ОК.
10. Переместите методом перетаскивания окно Аргументы функции, если оно закрывает нужные ячейки. Обратите внимание, что автоматически выбранный диапазон включает все ячейки с числовым содержанием, включая и ту, которая содержит сумму. Выделите правильный диапазон методом протягивания и нажмите клавишу ENTER.
11. Используя порядок действий, описанный в пп. 6–10, вычислите минимальное число в заданном наборе (функция МИН), максимальное число (МАКС), количество элементов в наборе (СЧЕТ).
12. Сохраните рабочую книгу book.xls.

Итог. Мы познакомились с некоторыми итоговыми функциями. Мы научились использовать итоговые функции для вычисления значений, характеризующих набор данных. Мы выяснили, как автоматически определяется диапазон значений, обрабатываемых функцией, и как изменить его вручную.

Упражнение 12.3. Подготовка и форматирование прайс-листа

1. Запустите программу Excel (Пуск ► Все программы ► Microsoft Office ► Microsoft Excel 2010) и откройте рабочую книгу book.xlsx.
2. Выберите щелчком на ярлычке неиспользуемый рабочий лист или создайте новый (кнопка Вставить лист в правой части панели ярлыков листов). Дважды щелкните на ярлычке нового листа и переименуйте его как Прейскурант.

3. В ячейку A1 введите текст **Прейскурант** и нажмите клавишу **ENTER**.
4. В ячейку A2 введите текст **Курс пересчета:** и нажмите клавишу **ENTER**. В ячейку B2 введите текст **1 у.е.=** и нажмите клавишу **ENTER**. В ячейку C2 введите текущий курс пересчета и нажмите клавишу **ENTER**.
5. В ячейку A3 введите текст **Наименование товара** и нажмите клавишу **ENTER**. В ячейку B3 введите текст **Цена (у.е.)** и нажмите клавишу **ENTER**. В ячейку C3 введите текст **Цена (руб.)** и нажмите клавишу **ENTER**.
6. В последующие ячейки столбца A введите названия товаров, включенных в **прейскурант**.
7. В соответствующие ячейки столбца B введите цены товаров в условных единицах.
8. В ячейку C4 введите формулу: **=B4*\$C\$2**, которая используется для пересчета цены из условных единиц в рубли.
9. Методом автозаполнения скопируйте формулы во все ячейки столбца C, которым соответствуют заполненные ячейки столбцов A и B. Почему при таком копировании получатся верные формулы?
10. Измените курс пересчета в ячейке C2. Обратите внимание, что все цены в рублях при этом обновляются автоматически.
11. Выделите методом протягивания диапазон A1:C1 и дайте команду контекстного меню **Формат ячеек**. На вкладке **Выравнивание** задайте выравнивание по горизонтали **По центру** и установите флажок **Объединение ячеек**.
12. На вкладке **Шрифт** задайте размер шрифта равный **14** пунктам и в списке **Начертание** выберите вариант **Полужирный**. Щелкните на кнопке **ОК**.
13. Щелкните правой кнопкой мыши на ячейке B2 и выберите в контекстном меню команду **Формат ячеек**. Задайте выравнивание по горизонтали **По правому краю** и щелкните на кнопке **ОК**.
14. Щелкните правой кнопкой мыши на ячейке C2 и выберите в контекстном меню команду **Формат ячеек**. Задайте выравнивание по горизонтали **По левому краю** и щелкните на кнопке **ОК**.
15. Выделите методом протягивания диапазон B2:C2. Щелкните на раскрывающейся кнопке рядом с кнопкой **Главная** ► **Шрифт** ► **Границы** и задайте для этих ячеек толстую внешнюю границу.
16. Дважды щелкните на границе между заголовками столбцов A и B, B и C, C и D. Обратите внимание, как при этом изменяется ширина столбцов A, B и C.
17. Посмотрите, устраивает ли вас полученный формат таблицы. Выполните команду **Файл** ► **Печать**, чтобы увидеть, как документ будет выглядеть при печати.
18. Щелкните на кнопке **Печать** и напечатайте документ.
19. Сохраните рабочую книгу **book.xlsx**.

Итог. Мы научились форматировать документ Excel. При этом мы использовали такие средства, как изменение ширины столбцов, объединение ячеек, управление выравниванием текста, создание рамок ячеек. Мы выяснили, что в готовом документе заданные и вычисленные ячейки отображаются одинаково. Мы познакомились с использованием средства предварительного просмотра и произвели печать документа.

Упражнение 12.4. Построение экспериментального графика

1. Запустите программу Excel (Пуск ► Все программы ► Microsoft Office ► Microsoft Excel 2010) и откройте рабочую книгу book.xlsx, созданную ранее.
2. Выберите щелчком на ярлычке неиспользуемый рабочий лист или создайте новый. Дважды щелкните на ярлычке листа и переименуйте его как Обработка эксперимента.
3. В столбец А, начиная с ячейки А1 по ячейку А20, введите произвольный набор значений независимой переменной.
4. В столбец В, начиная с ячейки В1 по ячейку В20, введите произвольный набор значений функции.
5. Методом протягивания выделите все заполненные ячейки столбцов А и В.
6. Выполните команду Вставка ► Диаграммы ► Точечная ► Точечная с гладкими кривыми и маркерами.
7. Стандартная диаграмма выбранного типа будет вставлена на рабочий лист. Через контекстное меню диаграммы вызовите окно Выбрать данные. В поле Ряд выберите Ряд 1 и нажмите на кнопку Изменить. В появившемся окне, в поле Имя, укажите Результаты измерений. Закройте открытые окна нажатиями на кнопки ОК.
8. Убедитесь, что заданное название ряда данных автоматически использовано как заголовок диаграммы. Выделите заголовок диаграммы и, убедившись, что он находится в режиме редактирования, замените его, введя заголовок Экспериментальные точки.
9. Выполните команду контекстного меню диаграммы Переместить диаграмму. В появившемся окне Перемещение диаграммы выберите На отдельном листе, введите имя для листа Диаграмма и нажмите ОК. Убедитесь, что диаграмма перемещена на отдельный лист.
10. Щелкните на построенной кривой, чтобы выделить ряд данных. Дайте команду контекстного меню ряда Формат ряда данных.
11. Откройте вкладку Линия, установите параметр Сплошная линия, откройте палитру Цвет и выберите красный цвет.
12. Откройте вкладку Тип линии, в списке Тип штриха выберите пунктир.
13. На вкладке Параметры маркера установите параметр Встроенный, выберите в списке Тип треугольный маркер.
14. Откройте вкладку Заливка маркера, установите параметр Сплошная заливка, в палитре Цвет выберите зеленый.
15. Щелкните на кнопке ОК, снимите выделение с ряда данных и посмотрите, как изменился вид графика.
16. Сохраните рабочую книгу.

Итог. Мы научились строить графики на основе данных, содержащихся на рабочем листе, настраивать формат диаграммы, задавать отображаемые данные и оформлять получающуюся диаграмму. Мы также узнали, как можно изменить формат готовой диаграммы.

Упражнение 12.5. Анализ данных с использованием метода наименьших квадратов

Задача. Для заданного набора пар значений независимой переменной и функции определить наилучшее линейное приближение в виде прямой с уравнением $y = ax + b$ и показательное приближение в виде линии с уравнением $y = b a^x$.

1. Запустите программу Excel (Пуск ► Все программы ► Microsoft Office ► Microsoft Excel 2010) и откройте рабочую книгу book.xlsx, созданную ранее.
 2. Щелчком на ярлычке выберите рабочий лист Обработка эксперимента.
 3. Сделайте ячейку C1 текущей и щелкните на кнопке Вставить функцию в строке формул.
 4. В окне мастера функций выберите категорию Ссылки и массивы и функцию ИНДЕКС. В новом диалоговом окне выберите первый вариант набора параметров.
 5. Установите текстовый курсор в первое поле для ввода параметров в окне Аргументы функции и выберите в раскрывающемся списке в строке формул пункт Другие функции.
 6. С помощью мастера функций выберите функцию ЛИНЕЙН категории Статистические.
 7. В качестве первого параметра функции ЛИНЕЙН выберите диапазон, содержащий значения функции (столбец В).
 8. В качестве второго параметра функции ЛИНЕЙН выберите диапазон, содержащий значения независимой переменной (столбец А).
 9. Переместите текстовый курсор в строке формул, чтобы он стоял на имени функции ИНДЕКС. В качестве второго параметра функции ИНДЕКС задайте число 1. Щелкните на кнопке ОК в окне Аргументы функции.
- Функция ЛИНЕЙН возвращает коэффициенты уравнения прямой в виде массива из двух элементов. С помощью функции ИНДЕКС выбирается нужный элемент.
10. Сделайте текущей ячейку D1. Повторите операции, описанные в пп. 3–9, чтобы в итоге в этой ячейке появилась формула: =ИНДЕКС(ЛИНЕЙН(В1:В20;А1:А20);2). Ее можно ввести и вручную (посимвольно). Теперь в ячейках С1 и D1 вычислены соответственно коэффициенты a и b уравнения наилучшей прямой.
 11. Сделайте текущей ячейку С2. Повторите операции, описанные в пп. 3–9, или введите вручную следующую формулу:

=ИНДЕКС(ЛГРФПРИБЛ(В1:В20;А1:А20);1)

12. Сделайте текущей ячейку D2. Повторите операции, описанные в пп. 3–9, или введите вручную следующую формулу:

=ИНДЕКС(ЛГРФПРИБЛ(В1:В20;А1:А20);2)

Теперь ячейки С2 и D2 содержат соответственно коэффициенты a и b уравнения наилучшего показательного приближения.

Для интерполяции или экстраполяции оптимальной кривой без явного определения ее параметров можно использовать функции ТЕНДЕНЦИЯ (для линейной зависимости) и РОСТ (для показательной зависимости).

13. Для построения наилучшей прямой другим способом дайте команду Данные ► Анализ ► Анализ данных.
14. Откроется одноименное диалоговое окно. В списке Инструменты анализа выберите пункт Регрессия, после чего щелкните на кнопке ОК.
15. В поле Входной интервал Y укажите методом протягивания диапазон, содержащий значения функции (столбец B).
16. В поле Входной интервал X укажите методом протягивания диапазон, содержащий значения независимой переменной (столбец A).
17. Установите переключатель Новый рабочий лист и задайте для него имя Результат расчета.
18. Щелкните на кнопке ОК и по окончании расчета откройте рабочий лист Результат расчета. Убедитесь, что вычисленные коэффициенты (см. ячейки B17 и B18) совпали с полученными первым методом.
19. Сохраните рабочую книгу book.xls.

Итог. Мы научились анализировать с помощью программы Excel экспериментальные данные с использованием метода наименьших квадратов. Мы применили для вычислений разные средства программы Excel. Мы получили информацию, необходимую для построения графиков нужных приближений.

Упражнение 12.6. Табулирование формул, построение графиков функций

Задача. Построить графики функций, коэффициенты которых определены в предыдущем упражнении.

1. Запустите программу Excel (Пуск ► Все программы ► Microsoft Office ► Microsoft Excel 2010) и откройте рабочую книгу book.xlsx.
2. Выберите щелчком на ярлычке рабочий лист Обработка эксперимента.
3. Так как программа Excel не позволяет непосредственно строить графики функций, заданных формулами, необходимо сначала табулировать формулу, то есть создать таблицу значений функций для заданных значений переменной.
4. В ячейку G1 введите формулу $=A1* \$C\$1 + \$D\$1$. Здесь абсолютные ссылки на ячейки C1 и D1 ведут к вычисленным методом наименьших квадратов коэффициентам уравнения прямой.
5. В ячейку H1 введите формулу $=\$D\$2* \$C\$2^A1$ для вычисления значения показательной функции.
6. Выделите ячейки G1 и H1 и путем протягивания заполните их формулами. Убедитесь, что при автозаполнении ячеек изменяется лишь ссылка на ячейку столбца A, который содержит независимую переменную функции.

7. Переключитесь на рабочий лист **Диаграмма**.
8. Из контекстного меню диаграммы выберите пункт **Выбрать данные**.
9. В группе **Элементы легенды** ряды и щелкните на кнопке **Добавить**. В поле **Имя** укажите: **Наилучшая прямая**. В поле **Значения X** укажите диапазон ячеек с данными в столбце **A**, а в поле **Значения Y** укажите диапазон ячеек в столбце **G**.
10. Еще раз щелкните на кнопке **Добавить**. В поле **Имя** укажите: **Показательная функция**. В поле **Значения X** укажите диапазон ячеек с данными в столбце **A**, а в поле **Значения Y** укажите диапазон ячеек в столбце **H**.
11. Щелкните на кнопке **ОК**, убедитесь в том, что диаграмма перестроена в соответствии с новыми настройками.
12. Сохраните рабочую книгу **book.xlsx**.

Итог. Мы научились создавать таблицу подстановки, содержащую значения заданных формул для нужных значений независимой переменной. Мы применили эту возможность программы Excel для построения графиков функций, заданных формулами. Мы также научились редактировать ранее построенную диаграмму, нанося на нее дополнительные графики.

Упражнение 12.7. Решение уравнений средствами программы Excel

Задача. Найти решение уравнения $x^3 - 3x^2 + x = -1$.

1. Запустите программу Excel (**Пуск** ► **Все программы** ► **Microsoft Office** ► **Microsoft Excel 2010**) и откройте рабочую книгу **book.xlsx**, созданную ранее.
2. Создайте новый рабочий лист, дважды щелкните на его ярлычке и присвойте ему имя **Уравнение**.
3. Занесите в ячейку **A1** значение 0.
4. Занесите в ячейку **B1** левую часть уравнения, используя в качестве независимой переменной ссылку на ячейку **A1**. Соответствующая формула может, например, иметь вид **=A1^3-3*A1^2+A1**.
5. Дайте команду **Данные** ► **Анализ** ► **Поиск решения**.
6. В поле **Оптимизировать целевую функцию** укажите **B1**, установите переключатель **До значения** и задайте значение **-1**, в поле **Изменяя ячейки переменных** укажите **A1**.
7. Щелкните на кнопке **Найти решение** и посмотрите на результат поиска, который отобразится на рабочем листе. Установите переключатель **Сохранить найденное решение** и щелкните на кнопке **ОК**, чтобы сохранить полученные значения ячеек, участвовавших в операции.
8. Сохраните рабочую книгу **book.xlsx**.

Итог. Мы научились численно решать с помощью программы Excel уравнения, содержащие одно неизвестное и задаваемые формулой. Похожим способом можно решать уравнения, содержащие несколько переменных, а также системы уравнений.

Упражнение 12.8. Решение задач оптимизации

Задача. Завод производит электронные приборы трех видов (прибор А, прибор В и прибор С), используя при сборке микросхемы трех типов (тип 1, тип 2 и тип 3). Расход микросхем задается следующей таблицей:

	Прибор А	Прибор В	Прибор С
Тип 1	2	5	1
Тип 2	2	0	4
Тип 3	2	1	1

Стоимость изготовленных приборов одинакова.

Ежедневно на склад завода поступает 400 микросхем типа 1 и по 500 микросхем типов 2 и 3. Каково оптимальное соотношение дневного производства приборов различного типа, если производственные мощности завода позволяют использовать запас поступивших микросхем полностью?

1. Запустите программу Excel (Пуск ► Все программы ► Microsoft Office ► Microsoft Excel 2010) и откройте рабочую книгу book.xlsx, созданную ранее.
2. Создайте новый рабочий лист, дважды щелкните на его ярлычке и присвойте ему имя Организация производства.
3. В ячейки A2, A3 и A4 занесите дневной запас комплектующих — числа 400, 500 и 500 соответственно.
4. В ячейки C1, D1 и E1 занесите нули — в дальнейшем значения этих ячеек будут подобраны автоматически.
5. В ячейках диапазона C2:E4 разместите таблицу расхода комплектующих.
6. В ячейках B2:B4 нужно указать формулы для расчета расхода комплектующих по типам. В ячейке B2 формула будет иметь вид $=\$C\$1*C2+\$D\$1*D2+\$E\$1*E2$, а остальные формулы можно получить методом автозаполнения (обратите внимание на использование абсолютных и относительных ссылок).
7. В ячейку F1 занесите формулу, вычисляющую общее число произведенных приборов: для этого выделите диапазон C1:E1 и щелкните на кнопке Автосумма на стандартной панели инструментов.
8. Дайте команду Данные ► Анализ ► Поиск решения — откроется диалоговое окно Параметры поиска решения.
9. В поле Оптимизировать целевую функцию укажите ячейку, содержащую оптимизируемое значение (F1). Установите переключатель Максимум (требуется максимальный объем производства).
10. В поле Изменяя ячейки переменных задайте диапазон подбираемых параметров — C1:E1.
11. Чтобы определить набор ограничений, щелкните на кнопке Добавить. В диалоговом окне Добавление ограничения в поле Ссылка на ячейку укажите диапазон B2:B4. В качестве условия задайте $\leq$. В поле Ограничение задайте диапазон

A2:A4. Это условие указывает, что дневной расход комплектующих не должен превосходить запасов. Щелкните на кнопке ОК.

12. Снова щелкните на кнопке **Добавить**. В поле **Ссылка на ячейку** укажите диапазон C1:E1. В качестве условия задайте $\geq$. В поле **Ограничение** задайте число 0. Это условие указывает, что число производимых приборов неотрицательно. Щелкните на кнопке ОК.
13. Снова щелкните на кнопке **Добавить**. В поле **Ссылка на ячейку** укажите диапазон C1:E1. В качестве условия выберите пункт **цел**. Это условие не позволяет производить доли приборов. Щелкните на кнопке ОК.
14. Щелкните на кнопке **Найти решение**. По завершении оптимизации откроется диалоговое окно **Результаты поиска решения**.
15. Установите переключатель **Сохранить найденное решение**, после чего щелкните на кнопке ОК.
16. Проанализируйте полученное решение. Кажется ли оно очевидным? Проверьте его оптимальность, экспериментируя со значениями ячеек C1:E1. Чтобы восстановить оптимальные значения, можно в любой момент повторить операцию поиска решения.
17. Сохраните рабочую книгу book.xlsx.

Итог. Мы узнали, как использовать программу Excel для решения сложных задач оптимизации. Мы научились формулировать условия задачи табличным образом, формировать ограничения, которым должно удовлетворять решение, и производить поиск оптимального набора переменных. Мы также выяснили, что даже для несложной задачи оптимизации найти оптимальное решение подбором практически невозможно.

Глава тринадцатая

Работа с базами данных

13.1. Основные понятия баз данных

Базы данных и системы управления базами данных

База данных — это организованная структура, предназначенная для хранения информации. Внимательный читатель, знающий из первой главы этого пособия о том, что данные и информация — понятия взаимосвязанные, но не тождественные, должен заметить некоторое несоответствие в этом определении. Его причины чисто исторические. В те годы, когда формировалось понятие баз данных, в них действительно хранились только данные. Однако сегодня большинство *систем управления базами данных (СУБД)* позволяют размещать в своих структурах не только данные, но и методы (то есть программный код), с помощью которых происходит взаимодействие с потребителем или с другими программно-аппаратными комплексами. Таким образом, мы можем говорить, что **в современных базах данных хранятся отнюдь не только данные, но и информация.**

Это утверждение легко пояснить, если, например, рассмотреть базу данных крупного банка. В ней есть все необходимые сведения о клиентах, об их адресах, кредитной истории, состоянии расчетных счетов, финансовых операциях и т. д. Доступ к этой базе имеется у достаточно большого количества сотрудников банка, но среди них вряд ли найдется такое лицо, которое имеет доступ *ко всей базе полностью* и при этом способно единолично вносить в нее произвольные изменения. Кроме данных, база содержит *методы и средства*, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

С понятием *базы данных* тесно связано понятие *системы управления базой данных*. Это комплекс программных средств, предназначенных для создания структуры новой базы, наполнения ее содержимым, редактирования содержимого и визуализации информации. Под *визуализацией информации* базы понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода или передача по каналам связи.

В мире существует множество систем управления базами данных. Несмотря на то что они могут по-разному работать с разными объектами и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных понятий. Это дает нам возможность рассмотреть одну систему и обобщить ее понятия, приемы и методы на весь класс СУБД. В качестве такого учебного объекта мы выберем СУБД Microsoft Access, входящую в пакет Microsoft Office наряду с рассмотренными ранее пакетами Microsoft Word и Microsoft Excel. В тех случаях, когда конкретные приемы опе-

раций зависят от используемой версии программы, мы будем опираться на последнюю версию Microsoft Access 2010, хотя в основном речь будет идти о таких обобщенных понятиях и методах, для которых различия между конкретными версиями программ второстепенны.

Структура простейшей базы данных

Сразу поясним, что если в базе нет никаких данных (*пустая база*), то это все равно полноценная база данных. Этот факт имеет методическое значение. Хотя данных в базе и нет, но информация в ней все-таки есть — это *структура базы*. Она определяет методы занесения данных и хранения их в базе. Простейший «некомпьютерный» вариант базы данных — деловой ежедневник, в котором каждому календарному дню выделено по странице. Даже если в нем не записано ни строки, он не перестает быть ежедневником, поскольку имеет структуру, четко отличающую его от записных книжек, рабочих тетрадей и прочей писчебумажной продукции.

Базы данных могут содержать различные объекты, но, забегая вперед, скажем, что основными объектами любой базы данных являются ее таблицы. Простейшая база данных имеет хотя бы одну таблицу. Соответственно, структура простейшей базы данных тождественно равна структуре ее таблицы.

Мы знаем, что структуру двумерной таблицы образуют столбцы и строки (рис. 13.1). Их аналогами в структуре простейшей базы данных являются *поля* и *записи*. Если записей в таблице пока нет, значит, ее структура образована только набором полей. Изменив состав полей базовой таблицы (или их свойства), мы изменяем структуру базы данных и, соответственно, получаем новую базу данных.

Записи		Поля	
Комплектующие			
Микропроцессор	Модель	Цена	Поставщик
Intel Core	i5-650	6 114,00р.	ООО "Самшит"
Intel Core	i5-670	11 271,00р.	ООО "Планета"
Intel Core	i5-750	7 026,00р.	ЗАО "Система"
Intel Core	i5-760	6 927,00р.	ООО "Кристина"
*			
Записи: 14 5 из 5 Нет фильтра Поиск			

Рис. 13.1. Простейшая таблица базы данных

Свойства полей базы данных

Поля базы данных не просто определяют структуру базы — они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. Ниже перечислены основные свойства полей таблиц баз данных на примере СУБД Microsoft Access.

- **Имя поля** — определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).

- ❑ Тип поля — определяет тип данных, которые могут содержаться в данном поле.
- ❑ Размер поля — определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.
- ❑ Формат поля — определяет способ форматирования данных в ячейках, принадлежащих полю.
- ❑ Маска ввода — определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных).
- ❑ Подпись — определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство Имя поля).
- ❑ Значение по умолчанию — то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).
- ❑ Условие на значение — ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).
- ❑ Сообщение об ошибке — текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности выполняется автоматически, если задано свойство Условие на значение).
- ❑ Обязательное поле — свойство, определяющее обязательность заполнения данного поля при наполнении базы.
- ❑ Пустые строки — свойство, разрешающее ввод пустых строковых данных (от свойства Обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например к текстовым).
- ❑ Индексированное поле — если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значения в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Здесь мы должны обратить особое внимание читателя на то, что поскольку в разных полях могут содержаться данные разного типа, то и свойства у полей могут различаться в зависимости от типа данных. Так, например, список вышеуказанных свойств полей относится в основном к полям текстового типа. Поля других типов могут иметь или не иметь эти свойства, но могут добавлять к ним и свои. Например, для данных, представляющих действительные числа, важным свойством является количество знаков после десятичной запятой. С другой стороны, для полей, используемых для хранения рисунков, звукозаписей, видеоклипов и других объектов *OLE*, большинство вышеуказанных свойств не имеет смысла.

Типы данных

С основными типами данных мы уже знакомы. Так, например, при изучении электронных таблиц Microsoft Excel мы видели, что они работают с тремя типами данных: текстами, числами и формулами. Таблицы баз данных, как правило, допу-

скают работу с гораздо большим количеством разных типов данных. Так, например, базы данных Microsoft Access работают со следующими типами данных (рис. 13.2).

Поле счетчика (заполняется автоматически, хранит данные числового типа)

Поле денежного типа

Поле типа «Дата»

№ п/п	Модель	Размер	Цена	Наличие	Ожидается	Примечание	Щелкните для добавления
1	Acer V193HQVb	19"	6 114,00р.	☒		Гарантия фирмы	
2	LG E2340S-PN	23"	7 431,00р.	☒		Гарантия фирмы	
3	Samsung SyncMaster B2430H	24"	9 545,00р.	☐	22.09.2010		
(No)				☐			

Текстовые поля

Поле логического типа

Рис. 13.2. Таблица с полями некоторых типов

- **Текстовый** — тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).
- **Поле Мемо** — специальный тип данных для хранения больших объемов текста. Физически текст не хранится в поле. Он хранится в другом месте базы данных, а в поле хранится указатель на него, но для пользователя такое разделение заметно не всегда.
- **Числовой** — тип данных для хранения действительных чисел.
- **Дата/время** — тип данных для хранения календарных дат и текущего времени.
- **Денежный** — тип данных для хранения денежных сумм. Теоретически, для их записи можно было бы пользоваться и полями числового типа, но для денежных сумм есть некоторые особенности (например, связанные с правилами округления), которые делают более удобным использование специального типа данных, а не настройку числового типа.
- **Счетчик** — специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование — для порядковой нумерации записей.
- **Логический** — тип для хранения логических данных (могут принимать только два значения, например Да или Нет).
- **Поле объекта OLE** — специальный тип данных, предназначенный для хранения объектов OLE, например мультимедийных. Реально, конечно, такие объекты в таблице не хранятся. Как и в случае полей МЕМО, они хранятся в другом месте внутренней структуры файла базы данных, а в таблице хранятся только указатели на них (иначе работа с таблицами была бы чрезвычайно замедленной).
- **Гиперссылка** — специальное поле для хранения адресов URL для веб-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.
- **Вложение** — предназначен для хранения в базе данных документов различных типов.

- ❑ **Вычисляемый** — содержит значение, вычисляемое на основе других данных.
- ❑ **Мастер подстановок** — это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

Безопасность баз данных

Базы данных — это тоже файлы, но работа с ними отличается от работы с файлами других типов, создаваемых прочими приложениями. Выше мы видели, что всю работу по обслуживанию файловой структуры берет на себя операционная система. Для баз данных предъявляются особые требования с точки зрения безопасности, поэтому в них реализован другой подход к сохранению данных.

При работе с обычными приложениями для сохранения данных мы выдаем соответствующую команду, задаем имя файла и доверяемся операционной системе. Если мы закроем файл, не сохранив его, то вся работа по созданию или редактированию файла пропадет безвозвратно.

Базы данных — это особые структуры. Информация, которая в них содержится, очень часто имеет общественную ценность. Нередко с одной и той же базой (например, с базой регистрации автомобилей в ГИБДД) работают тысячи людей по всей стране. От информации, которая содержится в некоторых базах, может зависеть благополучие множества людей. Поэтому целостность содержимого базы не может и не должна зависеть ни от конкретных действий некоего пользователя, забывшего сохранить файл перед выключением компьютера, ни от перебоев в электросети.

Проблема безопасности баз данных решается тем, что в СУБД для сохранения информации используется двойной подход. В части операций, как обычно, участвует операционная система компьютера, но некоторые операции сохранения происходят в обход операционной системы.

Операции изменения структуры базы данных, создания новых таблиц или иных объектов происходят при сохранении файла базы данных. Об этих операциях СУБД предупреждает пользователя. Это, так сказать, глобальные операции. Их никогда не проводят с базой данных, находящейся в коммерческой эксплуатации, — только с ее копией. В этом случае любые сбои в работе вычислительных систем не страшны.

С другой стороны, операции по изменению содержания данных, не затрагивающие структуру базы, максимально автоматизированы и выполняются без предупреждения. Если, работая с таблицей данных, мы что-то в ней меняем в составе данных, то изменения сохраняются немедленно и автоматически.

Обычно, решив отказаться от изменений в документе, его просто закрывают без сохранения и вновь открывают предыдущую копию. Этот прием работает почти во всех приложениях, но только не в СУБД. Все изменения, вносимые в таблицы базы, сохраняются на диске без нашего ведома, поэтому попытка закрыть базу «без сохранения» ничего не даст, так как все уже сохранено. Таким образом, редактируя таблицы баз данных, создавая новые записи и удаляя старые, мы как бы работаем с жестким диском напрямую, минуя операционную систему.

По указанным выше причинам нельзя заниматься учебными экспериментами на базах данных, находящихся в эксплуатации. Для этого следует создавать специальные учебные базы или выполнять копии структуры реальных баз (без фактического наполнения данными).

13.2. Формирование баз данных

Режимы работы с базами данных

Обычно с базами данных работают две категории исполнителей. Первая категория — *проектировщики*. Их задача состоит в разработке структуры таблиц базы данных и согласовании ее с заказчиком. Кроме таблиц проектировщики разрабатывают и другие объекты базы данных, предназначенные, с одной стороны, для автоматизации работы с базой, а с другой стороны — для ограничения функциональных возможностей работы с базой (если это необходимо из соображений безопасности). Проектировщики не наполняют базу конкретными данными (заказчик может считать их конфиденциальными и не предоставлять посторонним лицам). Исключение составляет экспериментальное наполнение модельными данными на этапе отладки объектов базы.

Вторая категория исполнителей, работающих с базами данных, — *пользователи*. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием. В общем случае пользователи не имеют средств доступа к управлению структурой базы — только к данным, да и то не ко всем, а к тем, работа с которыми предусмотрена на конкретном рабочем месте.

Соответственно, система управления базами данных имеет два режима работы: *проектировочный* и *пользовательский*. Первый режим предназначен для создания или изменения структуры базы и создания ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

Объекты базы данных

Мы уже упомянули о том, что кроме таблиц база данных может содержать и другие типы объектов. Привести полную классификацию возможных объектов баз данных затруднительно, поскольку каждая система управления базами данных может реализовать свои типы объектов. Однако основные типы объектов мы можем рассмотреть на примере СУБД Microsoft Access. В версии Microsoft Access 2010 эта СУБД позволяет создавать и использовать объекты семи различных типов.

Таблицы. Как мы уже говорили, это основные объекты любой базы данных. В первых, в таблицах хранятся все данные, имеющиеся в базе, а во вторых, таблицы хранят и структуру базы (поля, их типы и свойства).

Запросы. Эти объекты служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С помощью запросов выполняют такие операции, как отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять преобразование данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импорти-

рованными из других источников, выполнять простейшие вычисления в таблицах и многое другое.

Начинающие пользователи не сразу понимают роль запросов, поскольку все те же операции можно делать и с таблицами. Да, действительно, это так, но есть соображения удобства (в первую очередь быстродействия) и соображения безопасности.

Из соображений безопасности, чем меньше доступа к базовым таблицам имеют конечные пользователи, тем лучше. Во-первых, снижается риск того, что неумелыми действиями они повредят данные в таблицах. Во-вторых, предоставив разным пользователям разные запросы, можно эффективно разграничить их доступ к данным в строгом соответствии с кругом персональных обязанностей. В банках, например, одни сотрудники имеют доступ к таблицам данных о клиентах, другие — к их расчетным счетам, третьи — к таблицам активов банка. Если и есть специальные службы, имеющие доступ ко всем информационным ресурсам банка (с целью контроля и анализа), то они лишены средств для внесения изменений — все сделано так, чтобы один человек не мог совершить фиктивную операцию, независимо от того, какую должность он занимает. В базе данных, имеющей правильно организованную структуру, для совершения противоправных действий необходим сговор нескольких участников, а такие действия пресекаются не программными, а традиционными средствами обеспечения безопасности.

Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную *результатирующую таблицу*. Если хотят подчеркнуть факт «временности» этой таблицы, то ее еще называют *моментальным снимком*. Когда мы работаем с основными таблицами базы, мы физически имеем дело с жестким диском, то есть с очень медленным устройством (напомним, что это связано с особенностью сохранения данных, описанной выше). Когда же на основании запроса мы получаем результирующую таблицу, то имеем дело с электронной таблицей, не имеющей аналога на жестком диске, — это только *образ* отобранных полей и записей. Разумеется, работа с «образом» происходит гораздо быстрее и эффективнее — это еще одно основание для того, чтобы широко использовать запросы.

Когда в главе 1 мы обсуждали основные структуры данных, то отметили, что недостатком упорядоченных табличных структур является сложность их обновления, поскольку при внесении новых записей нарушается упорядоченность — приходится переделывать всю таблицу. В системах управления базами данных и эта проблема решается благодаря запросам.

Основной принцип состоит в том, что от базовых таблиц никакой упорядоченности не требуется. Все записи в основные таблицы вносятся только в *естественном* порядке по мере их поступления, то есть в неупорядоченном виде. Если же пользователю надо видеть данные, отсортированные или отфильтрованные по тому или иному принципу, он просто использует соответствующий запрос (рис. 13.3). Если нужного запроса нет, он обращается к проектировщику и просит его такой запрос сделать и предоставить.

Базовая таблица

Комплектующие	Микропроцессор	Модель	Цена	Поставщик
	Intel Core	i5-650	6 114,00р.	ООО "Самшит"
	Intel Core	i5-670	11 271,00р.	ООО "Планета"
	Intel Core	i5-750	7 026,00р.	ЗАО "Система"
	Intel Core	i5-760	6 927,00р.	ООО "Кристина"
	AMD Phenom II X2	545	3 054,00р.	ООО "Планета"
	AMD Phenom II X2	550	3 546,00р.	ЗАО "Система"
Записи: 7 из 7				

Поставщики процессоров

Поставщик	Микропроцессор	Модель
ООО "Самшит"	Intel Core	i5-650
ООО "Планета"	Intel Core	i5-670
ЗАО "Система"	Intel Core	i5-750
ООО "Кристина"	Intel Core	i5-760
ООО "Планета"	AMD Phenom II X2	545
ЗАО "Система"	AMD Phenom II X2	550
Записи: 7 из 7		

Результирующая таблица

Цена на процессоры

Модель	Цена
i5-650	6 114,00р.
i5-670	11 271,00р.
i5-750	7 026,00р.
i5-760	6 927,00р.
545	3 054,00р.
550	3 546,00р.
Записи: 1 из 6	

Результирующая таблица

Рис. 13.3. Два запроса, сформированные на основе одной таблицы

Формы. Если запросы — это специальные средства для отбора и анализа данных, то формы — это средства для ввода данных. Смысл их тот же — предоставить пользователю средства для заполнения только тех полей, которые ему заполнять положено. Одновременно с этим в форме можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и пр.) для автоматизации ввода. Преимущества форм раскрываются особенно наглядно, когда происходит ввод данных с заполненных бланков. В этом случае форму делают графическими средствами так, чтобы она повторяла оформление бланка, — это заметно упрощает работу наборщика, снижает его утомление и предотвращает появление печатных ошибок. На сопроводительном рисунке приведен пример простейшей формы для ввода данных.

С помощью форм данные можно не только вводить, но и отображать. Запросы тоже отображают данные, но делают это в виде результирующей таблицы, не имеющей почти никаких средств оформления. При выводе данных с помощью форм можно применять специальные средства оформления (рис. 13.4).

Отчеты. По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на печатающее устройство (принтер). В связи с этим отчеты отличаются тем, что в них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов (верхний и нижний колонтитулы, номера страниц, служебная информация о времени создания отчета и т. п.) (рис. 13.5).

Рис. 13.4. Форма для оформленного вывода данных

Микропроцессор	Модель	Цена	Поставщик
Intel Core	i5-650	6 114,00р.	ООО "Самшит"
Intel Core	i5-670	11 271,00р.	ООО "Планета"
Intel Core	i5-750	7 026,00р.	ЗАО "Система"
Intel Core	i5-760	6 927,00р.	ООО "Кристина"
AMD Phenom II X2	545	3 054,00р.	ООО "Планета"
AMD Phenom II X2	550	3 546,00р.	ЗАО "Система"

Рис. 13.5. Пример простейшего отчета

Макросы и модули. Эти категории объектов предназначены как для автоматизации повторяющихся операций при работе с системой управления базами данных, так и для создания новых функций путем программирования. В СУБД Microsoft Access *макросы* состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой. *Модули* создаются средствами внешнего языка программирования, в данном случае языка Visual Basic for Applications. Это одно из средств, с помощью которых разработчик базы может заложить в нее нестандартные функциональные возможности, удовлетворить специфические требования заказчика, повысить быстродействие системы управления, а также уровень ее защищенности.

Проектирование базы данных

Мы лишь рассмотрели основные понятия баз данных и еще не знакомы ни с одной системой управления базами данных, но тем не менее поднимаем вопрос о про-

ектировании базы. Это не случайно. Методически правильно начинать работу с карандашом и листом бумаги в руках, не используя компьютер. На данном этапе он просто не нужен. Неоптимальные решения и прямые ошибки, заложенные на этапе проектирования, впоследствии очень трудно устраняются, поэтому этот этап является основополагающим.

Разработка технического задания. Техническое задание на проектирование базы данных должен предоставить заказчик. Однако для этого он должен владеть соответствующей терминологией и знать, хотя бы в общих чертах, технические возможности основных систем управления базами данных. К сожалению, на практике такое положение встречается не всегда. Поэтому обычно используют следующие подходы:

- ☐ демонстрируют заказчику работу аналогичной базы данных, после чего согласовывают спецификацию отличий;
- ☐ если аналога нет, выясняют круг задач и потребностей заказчика, после чего помогают ему подготовить техническое задание.

При подготовке технического задания составляют:

- ☐ список исходных данных, с которыми работает заказчик;
- ☐ список выходных данных, которые необходимы заказчику для управления структурой своего предприятия;
- ☐ список выходных данных, которые не являются необходимыми для заказчика, но которые он должен предоставлять в другие организации (в вышестоящие структуры, в органы статистического учета, прочие административные и контролирующие организации).

При этом очень важно не ограничиваться взаимодействием с головным подразделением заказчика, а провести обсуждение со всеми службами и подразделениями, которые могут оказаться поставщиками данных в базу или их потребителями. Так, например, при подготовке базы данных для учета абитуриентов и студентов в высшем учебном заведении необходимо не только изучить документооборот ректората и всех деканатов, но и понять, что хотели бы получить от базы данных службы. Следует изучить работу подразделений, распределяющих учебную нагрузку преподавателей, отвечающих за распределение аудиторного фонда, за проживание студентов в общежитии и др. В расчет должны приниматься и такие службы, как библиотека, отдел кадров и пр. В любой момент может выясниться, например, что администрация библиотеки должна периодически поставлять кому-то отчеты, характеризующие читательскую активность студентов в зависимости от пола, возраста и социального положения. К возможным пожеланиям заказчика следует готовиться на этапе проектирования, до создания базы.

Разработка схемы данных

Выяснив основную часть данных, которые заказчик потребляет или предоставляет, можно приступить к созданию структуры базы, то есть структуры ее основных таблиц.

1. Работа начинается с составления генерального списка полей — он может насчитывать десятки и даже сотни позиций.

- В соответствии с типом данных, размещаемых в каждом поле, определяют наиболее подходящий тип для каждого поля.
- Далее распределяют поля генерального списка по базовым таблицам. На первом этапе распределение производят по функциональному признаку. Цель — обеспечить, чтобы ввод данных в одну таблицу производился, по возможности, в рамках одного подразделения, а еще лучше — на одном рабочем месте.

Наметив столько таблиц, сколько подразделений охватывает база данных, приступают к дальнейшему делению таблиц. Критерием необходимости деления является факт множественного повтора данных в соседних записях. На рис. 13.6 показана таблица, у которой в поле **Адрес** наблюдается повтор данных. Это явное свидетельство того, что таблицу надо поделить на две взаимосвязанные таблицы.

Микропроцессор	Модель	Цена	Поставщик	Адрес
Intel Core	i5-650	6 114,00р.	ООО "Самшит"	123456, Москва, ул. Индустриальная, д. 15
Intel Core	i5-670	11 271,00р.	ООО "Планета"	122678, Москва, ул. Промышленная, д. 45
Intel Core	i5-750	7 026,00р.	ЗАО "Система"	111222, Москва, Заводской проспект, д. 23
Intel Core	i5-760	6 927,00р.	ООО "Кристина"	124345, Москва, Технический проезд, д. 3
AMD Phenom II X2	545	3 054,00р.	ООО "Планета"	122678, Москва, ул. Промышленная, д. 45
AMD Phenom II X2	550	3 546,00р.	ЗАО "Система"	111222, Москва, Заводской проспект, д. 23

Микропроцессор	Модель	Цена	Поставщик
Intel Core	i5-650	6 114,00р.	ООО "Самшит"
Intel Core	i5-670	11 271,00р.	ООО "Планета"
Intel Core	i5-750	7 026,00р.	ЗАО "Система"
Intel Core	i5-760	6 927,00р.	ООО "Кристина"
AMD Phenom II X2	545	3 054,00р.	ООО "Планета"
AMD Phenom II X2	550	3 546,00р.	ЗАО "Система"

Наименование	Индекс	Город	Адрес	Телефон
ЗАО "Система"	111222	Москва	Заводской проспект, д. 23	553-45-67
ООО "Кристина"	124345	Москва	Технический проезд, д. 3	455-55-34
ООО "Планета"	122678	Москва	ул. Промышленная, д. 45	345-54-34
ООО "Самшит"	123456	Москва	ул. Индустриальная, д. 15	123-45-67

Рис. 13.6. Если данные в поле начинают повторяться, это признак того, что таблицу следует поделить

- В каждой из таблиц намечают *ключевое поле*. В качестве такового выбирают поле, данные в котором повторяться не могут. Например, для таблицы данных

о студентах таким полем может служить индивидуальный шифр студента. Для таблицы, в которой содержатся расписания занятий, такого поля можно и не найти, но его можно создать искусственным комбинированием полей «Время занятия» и «Номер аудитории». Эта комбинация неповторима, так как в одной аудитории в одно и то же время не принято проводить два различных занятия.

Если в таблице вообще нет никаких полей, которые можно было бы использовать как ключевые, всегда можно ввести дополнительное поле типа Счетчик — оно не может содержать повторяющихся данных по определению.

5. С помощью карандаша и бумаги расчерчивают связи между таблицами. На рис. 13.7 показан пример взаимосвязи между группой таблиц, составляющих одну базу данных. Такой чертеж называется *схемой данных*.

Существует несколько типов возможных связей между таблицами. Наиболее распространенными являются связи «один ко многим» и «один к одному». Связь между таблицами организуется на основе общего поля, причем в одной из таблиц оно обязательно должно быть ключевым, то есть на стороне «один» должно выступать ключевое поле, содержащее уникальные, неповторяющиеся значения. Значения на стороне «многие» могут повторяться.

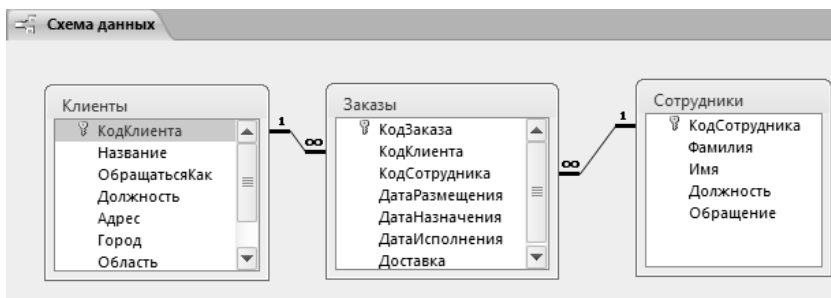


Рис. 13.7. Схема связей между таблицами

Рассмотрим таблицу Клиенты (см. рис. 13.7). Здесь поле КодКлиента является ключевым. Это понятно, поскольку у каждого клиента должен быть свой уникальный код, идентифицирующий его однозначно. Если мы рассмотрим таблицу Заказы, то увидим, что в ней код клиента не может быть уникальным, поскольку каждый клиент мог сделать сколь угодно много заказов. На схеме данных эти поля соединены *линией связи*. С одной стороны, эта линия маркирована знаком «1», с другой стороны — значком «бесконечность». Это графический метод изображения связи «один ко многим».

Ключевым полем в таблице заказов является КодЗаказа — он однозначно идентифицирует, кто, когда, что заказал и на какую сумму. Здесь же можно узнать, какой сотрудник принял заказ к исполнению. Поскольку один сотрудник может принять множество заказов, поле КодСотрудника в таблице заказов не является ни уникальным, ни ключевым, зато в таблице Сотрудники это поле уникально.

Про подобные таблицы говорят, что они связаны *реляционными отношениями*. Соответственно, системы управления, способные работать со связанными

таблицами, называют *системами управления реляционными базами данных*, а схему данных в технической литературе могут называть *схемой реляционных отношений*.

6. Разработкой схемы данных заканчивается «бумажный» этап работы над техническим предложением. Эту схему можно согласовать с заказчиком, после чего приступить к непосредственному созданию базы данных.

Следует помнить, что по ходу разработки проекта заказчику непременно будут приходить в голову новые идеи. На всех этапах проектирования он стремится охватить единой системой все новые и новые подразделения и службы предприятия. Возможность гибкого исполнения его пожеланий во многом определяется квалификацией разработчика базы данных. Если схема данных составлена правильно, подключать к базе новые таблицы нетрудно. Если структура базы нерациональна, разработчик может испытать серьезные трудности и войти в противоречия с заказчиком.

Противоречия исполнителя с заказчиком всегда свидетельствуют о недостаточной квалификации исполнителя. Именно поэтому этап предварительного проектирования базы данных следует считать основным. От его успеха зависит, насколько база данных станет удобной и будут ли с ней работать пользователи. Если отмечается, что пользователи базы «саботируют» ее эксплуатацию и предпочитают работать традиционными методами, это говорит не о низкой квалификации пользователей, а о недостаточной квалификации разработчика базы.

На этом этапе завершается предварительное проектирование базы данных, и на следующем этапе начинается ее непосредственная разработка. С этого момента следует начать работу с системой управления базами данных. В нашем примере мы рассмотрим СУБД Microsoft Access 2010.

13.3. Работа с СУБД Microsoft Access 2010

Общие замечания

Ниже мы рассмотрим, как в программе Microsoft Access 2010 реализованы средства разработки основных объектов базы данных, и в упражнениях познакомимся с конкретными приемами работы. Однако, прежде чем приступить к освоению системы, следует учесть ряд важных замечаний, связанных с особенностями ее автоматизации.

СУБД Microsoft Access 2010 предоставляет несколько средств создания каждого из основных объектов базы. Эти средства можно классифицировать как:

- ☐ ручные (разработка объектов в режиме Конструктора);
- ☐ автоматизированные (разработка с помощью программ-мастеров);
- ☐ автоматические — средства ускоренной разработки простейших объектов.

Соотношения между этими средствами понятны: ручные средства являются наиболее трудоемкими, но обеспечивают максимальную гибкость; автоматизированные и автоматические средства являются наиболее производительными, но и наименее гибкими. Методической особенностью изучения программы Microsoft

Access является тот факт, что в учебных целях для создания разных объектов целесообразно пользоваться разными средствами.

1. При разработке учебных таблиц и запросов рекомендуется использовать ручные средства — работать в режиме Конструктора. Использование мастеров ускоряет работу, но не способствует освоению понятий и методов.
2. При разработке учебных форм, отчетов и страниц доступа, наоборот, лучше пользоваться автоматизированными средствами, предоставляемыми мастерами. Это связано с тем, что для данных объектов большую роль играет внешний вид. Дизайн этих объектов весьма трудоемок, поэтому его лучше поручить программе, а учащемуся сосредоточиться на содержательной части работы.
3. Разработку макросов и модулей в данном пособии мы не рассматриваем. Эти средства ориентированы на профессиональных разработчиков баз данных, поэтому в рамках общетехнического курса «Информатики» для них недостаточно места.

Работа с таблицами

Создание таблиц. Создание объектов начинается с вкладки ленты Создание или с области навигации. На вкладке расположены элементы управления для вызова объектов, работая с областью навигации можно создавать и модифицировать объекты, пользуясь контекстным меню (рис. 13.8). Для того чтобы создать таблицу, можно воспользоваться командой Создание ► Таблицы ► Конструктор таблиц.

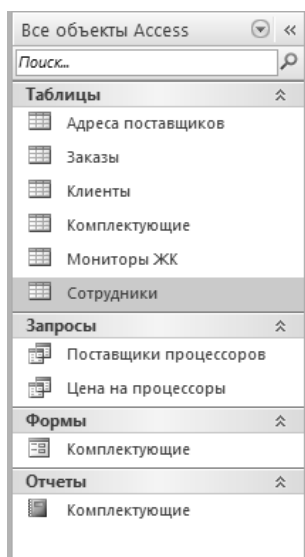


Рис. 13.8. Область навигации

После того как таблица создана, ее можно будет увидеть в области навигации, там она будет представлена в списке таблиц и других объектов базы данных. Окно

таблицы, в данном случае, это окно Конструктора таблиц (рис. 13.9), представлено в правой части окна программы.

То, что мы видим в этом режиме, фактически является графическим бланком для создания и редактирования структуры таблиц. В первом столбце вводят имена полей.

Если свойство Подпись для поля не задано, то Имя поля станет одновременно и именем столбца будущей таблицы. Тип для каждого поля выбирают из раскрывающегося списка, открываемого кнопкой выбора типа данных (см. рис. 13.9). Эта кнопка — скрытый элемент управления. Она отображается только после щелчка на поле бланка. Это надо иметь в виду — в Microsoft Access очень много таких скрытых элементов управления, которые не отображаются, пока ввод данных не начат.

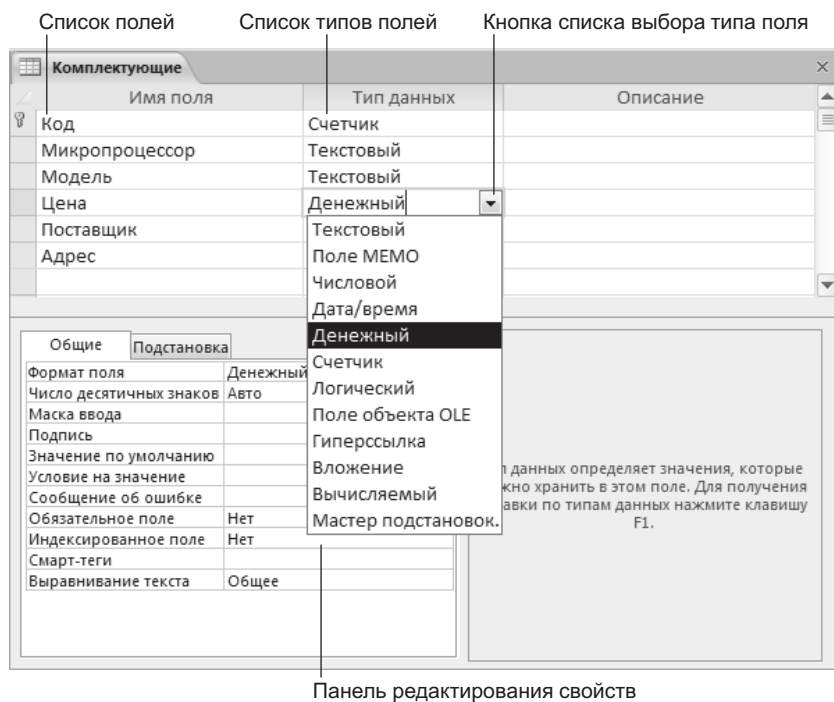


Рис. 13.9. Проектирование структуры таблицы

При изучении приемов работы с программой Microsoft Access целесообразно специально «прощелкивать» пустые поля ее бланков левой кнопкой мыши в поисках «скрытых» элементов управления.

Нижняя часть бланка содержит список свойств поля, выделенного в верхней части. Некоторые из свойств уже заданы по умолчанию. Свойства полей не являются обязательными. Их можно настраивать по желанию, а можно и не трогать.

При создании таблицы целесообразно (хотя и не обязательно) задать ключевое поле. Это поможет впоследствии, при организации связей между таблицами. Для задания ключевого поля достаточно щелкнуть на его имени правой кнопкой мыши и в открывшемся контекстном меню выбрать пункт Ключевое поле.

Выше мы говорили о том, что если первичный ключ необходим для связи с другими таблицами, но ни одно из полей не является уникальным, то первичный ключ можно создать на базе двух (или более полей). Эта операция выполняется точно так же, через контекстное меню, надо только уметь выделить сразу несколько полей. Групповое выделение выполняют при нажатой клавише **SHIFT** щелчками на квадратных маркерах слева от имен полей.

Закончив создание структуры таблицы, бланк закрывают (при этом система выдает запрос на сохранение таблицы), после чего дают таблице имя, и с этого момента она доступна в числе прочих таблиц в окне навигации. Оттуда ее и можно открыть в случае необходимости.

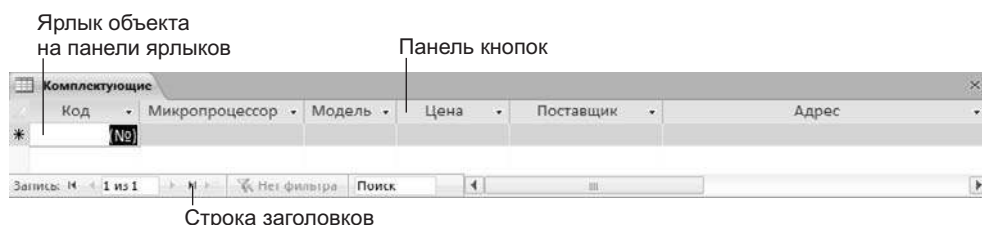


Рис. 13.10. Пример новой таблицы

Созданную таблицу открывают в окне навигации двойным щелчком на ее значке. Новая таблица не имеет записей — только названия столбцов, характеризующие структуру таблицы (рис. 13.10). Заполнение таблицы данными производится обычным порядком. Курсор ввода устанавливается в нужную ячейку указателем мыши. Переход к следующей ячейке можно выполнить клавишей **ТАВ**. Переход к очередной записи выполняется после заполнения последней ячейки.

В нижней части таблицы расположена **Панель кнопок перехода**. Ее элементами управления удобно пользоваться при навигации по таблице, имеющей большое число записей.

Начинающим пользователям Microsoft Access доставляет неудобство тот факт, что данные не всегда умещаются в ячейках таблицы. Шириной столбцов можно управлять методом перетаскивания их границ. Удобно использовать автоматическое форматирование столбцов «по содержимому». Для этого надо установить указатель мыши на границу между столбцами (в строке заголовков столбцов), дождаться, когда указатель сменит форму, и выполнить двойной щелчок. Это общесистемный прием *Windows* и им можно пользоваться в данной программе, как и во многих других.

После наполнения таблицы данными сохранять их не надо — все сохраняется автоматически. Однако если при работе с таблицей произошло редактирование ее макета (например, изменялась ширина столбцов), СУБД попросит подтвердить сохранение этих изменений.

Если возникнет необходимость изменить структуру таблицы (состав полей или их свойства), таблицу надо открыть в режиме Конструктора. Для этого ее следует выделить в окне области навигации и выполнить команду **Главная ► Режимы ► Режим ► Конструктор** или выполнить соответствующую команду контекстного меню таблицы.

Если на этапе проектирования базы данных была четко разработана структура таблиц, то создание таблиц с помощью Конструктора происходит очень быстро и эффективно. Даже без использования автоматизированных средств создание основы для достаточно крупных проектов происходит в считанные минуты — это ценное свойство СУБД Microsoft Access, но оно реализуется при обязательном условии тщательной предварительной подготовки.

Создание межтабличных связей. Если структура базы данных продумана заранее, а связи между таблицами намечены, то создание реляционных отношений между таблицами выполняется очень просто. Вся необходимая работа происходит в специальном окне **Схема данных** и выполняется с помощью мыши. Окно **Схема данных** открывают командой **Работа с базами данных ► Отношения ► Схема данных**. Порядок создания межтабличных связей рассмотрен подробно в упражнении 13.2.

Образовавшаяся межтабличная связь отображается в окне **Схема данных** в виде линии, соединяющей два поля разных таблиц. При этом одна из таблиц считается *главной*, а другая — *связанной*. Главная — это та таблица, которая участвует в связи своим ключевым полем (название этого поля на схеме данных отображается полужирным шрифтом).

Здесь мы подходим к важному вопросу: «А зачем вообще нужна связь между таблицами?» У связи два основных назначения. Первое — обеспечение целостности данных, а второе — автоматизация задач обслуживания базы. Представим себе, что в таблице **Клиенты**, где каждый клиент уникален, кто-то удалит запись для одного из клиентов, но не сделает этого в таблице **Заказы**. Получится, что согласно таблице **Заказы** некто, не имеющий ни имени, ни адреса, а только абстрактный код, делал заказы. Узнать по коду, кто же это был на самом деле, будет невозможно — произошло нарушение целостности данных.

В данном случае владелец базы может применить три подхода: либо вообще ничего не делать для защиты целостности данных, либо запретить удаление данных из ключевых полей главных таблиц, либо разрешить его, но при этом адекватно обработать и связанные таблицы. Вручную сделать это чрезвычайно трудно, поэтому и нужны средства автоматизации.

Связь между таблицами позволяет:

- ☐ либо исключить возможность удаления или изменения данных в ключевом поле главной таблицы, если с этим полем связаны какие-либо поля других таблиц;
- ☐ либо сделать так, что при удалении (или изменении) данных в ключевом поле главной таблицы автоматически (и абсолютно корректно) произойдет удаление или изменение соответствующих данных в полях связанных таблиц.

Для настройки свойств связи надо в окне **Схема данных** выделить линию, соединяющую поля двух таблиц, щелкнуть на ней правой кнопкой мыши и открыть контекстное меню связи, после чего выбрать в нем пункт **Изменить связь** — откроется диалоговое окно **Изменение связей** (рис. 13.11). В нем показаны названия связанных таблиц и имена полей, участвующих в связи (здесь же их можно изменить), а также приведены элементы управления для обеспечения условий целостности данных.

Если установлен только флажок **Обеспечение целостности данных**, то удалять данные из ключевого поля главной таблицы нельзя. Если вместе с ним включены

флажки Каскадное обновление связанных полей и Каскадное удаление связанных записей, то соответственно операции редактирования и удаления данных в ключевом поле главной таблицы разрешены, но сопровождаются автоматическими изменениями в связанной таблице.

Таким образом, смысл создания реляционных связей между таблицами состоит, с одной стороны, в защите данных, а с другой стороны — в автоматизации внесения изменений сразу в несколько таблиц при изменениях в одной таблице.

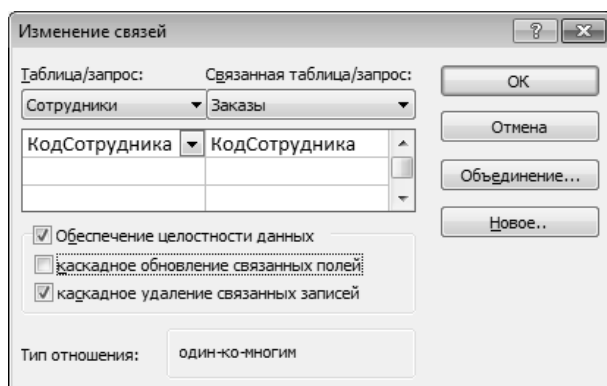


Рис. 13.11. Средство настройки межтабличной связи

Работа с запросами

Если структура базы данных предприятия хорошо продумана, то исполнители, работающие с базой, должны навсегда забыть о том, что в базе есть таблицы, а еще лучше, если они об этом вообще ничего не знают. Таблицы — слишком ценные объекты базы, чтобы с ними имел дело кто-либо, кроме разработчика базы.

Если исполнителю надо получить данные из базы, он должен использовать специальные объекты — запросы. Все необходимые запросы разработчик базы должен подготовить заранее. Если запрос подготовлен, надо найти его в области навигации, выбрать и открыть двойным щелчком на значке — откроется *результатирующая таблица*, в которой исполнитель найдет то, что его интересует.

В общем случае, результирующая таблица может не соответствовать ни одной из базовых таблиц базы данных. Ее поля могут представлять набор из полей разных таблиц, а ее записи могут содержать отфильтрованные и отсортированные записи таблиц, на основе которых формировался запрос. Лишь в тех случаях, когда исполнитель не находит нужных данных в результирующей таблице, возникает необходимость готовить новый запрос — это задача разработчика базы.

В учебных целях запросы лучше готовить вручную, с помощью Конструктора. Как и в случае с таблицами, для этого есть команда Создание ► Запросы ► Конструктор запросов. Команда открывает специальный бланк, называемый *бланком запроса по образцу*. За этим длинным названием скрывается тот приятный факт, что хотя запросы к таблицам баз данных пишутся на специальном языке программирования — SQL, пользователям Microsoft Access изучать его не обязательно, а большинство операций можно выполнить щелчками кнопок мыши и приемом перетаскивания в бланке.

Бланк запроса по образцу представлен на рис. 13.12. Как видно, он состоит из двух областей. В верхней отображается структура таблиц, к которым запрос адресован, а нижняя область разбита на столбцы — по одному столбцу на каждое поле будущей результирующей таблицы.

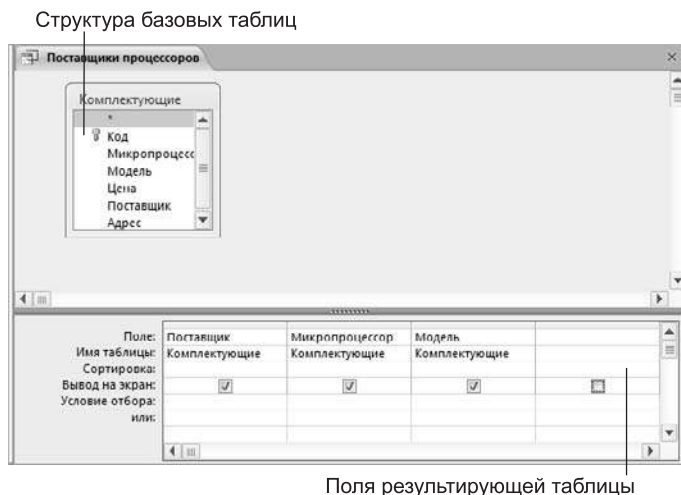


Рис. 13.12. Бланк запроса по образцу

Идея формирования запроса по образцу чрезвычайно проста. С помощью контекстного меню на верхней половине бланка открывают те таблицы, к которым обращен запрос. Затем в них щелкают двойными щелчками на названиях тех полей, которые должны войти в результирующую таблицу. При этом автоматически заполняются столбцы в нижней части бланка. Сформировав структуру запроса, его закрывают, дают ему имя и в дальнейшем запускают двойным щелчком на значке в области навигации.

Порядок действий, рассмотренный выше, позволяет создать простейший запрос, называемый *запросом на выборку*. Он позволяет выбрать данные из полей таблиц, на основе которых запрос сформирован.

Упорядочение записей в результирующей таблице. Если необходимо, чтобы данные, отобранные в результате работы запроса на выборку, были упорядочены по какому-либо полю, применяют сортировку. В нижней части бланка имеется специальная строка Сортировка. При щелчке на этой строке открывается кнопка раскрывающегося списка, в котором можно выбрать метод сортировки: по возрастанию или по убыванию. В результирующей таблице данные будут отсортированы по тому полю, для которого задан порядок сортировки.

Возможна многоуровневая сортировка — сразу по нескольким полям. В этом случае данные сначала сортируются по тому полю, которое в бланке запроса по образцу находится левее, затем по следующему полю, для которого включена сортировка, и так далее слева направо. Соответственно, при формировании запроса надо располагать поля результирующей таблицы не как попало, а с учетом будущей сортировки. В крайнем случае, если запрос уже сформирован и надо изменить порядок следования столбцов, пользуются следующим приемом:

- ❑ выделяют столбец щелчком на его заголовке (кнопку мыши отпускают);
- ❑ еще раз щелкают на заголовке уже выделенного столбца (но кнопку не отпускают);
- ❑ перетаскивают столбец в другое место.

Управление отображением данных в результирующей таблице. В нижней части бланка запроса по образцу имеется строка Вывод на экран. По умолчанию предполагается, что все поля, включенные в запрос, должны выводиться на экран, но это не всегда целесообразно. Например, бывают случаи, когда некоторое поле необходимо включить в запрос, например, потому, что оно является полем сортировки, но в то же время нежелательно, чтобы пользователь базы видел его содержание. В таких случаях отображение содержимого на экране подавляют сбросом флажка Вывод на экран. Примером может быть запрос на вывод списка сотрудников предприятия, отсортированный по количеству дней, пропущенных по болезни. Он позволит каждому оценить свое положение в этом списке, но не позволит точно узнать, кто сколько дней болел.

Использование условия отбора. Дополнительным средством, обеспечивающим отбор данных по заданному критерию, является так называемое Условие отбора. Соответствующая строка имеется в нижней части бланка запроса по образцу. Для каждого поля в этой строке можно задать индивидуальное условие. На рис. 13.13 представлен пример, в котором мы хотим отобрать из прейскуранта торговой фирмы микропроцессоры, стоящие более 6000 рублей.

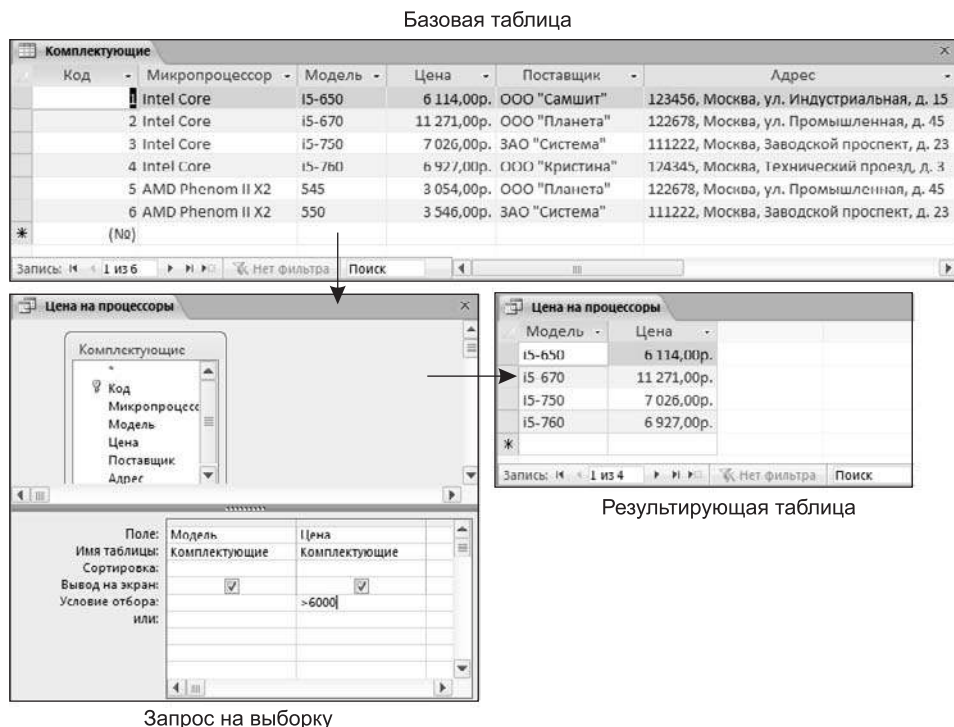


Рис. 13.13. Работа Условия отбора

Другие виды запросов. Мы рассмотрели запросы на выборку. Это самые простые и в то же время наиболее распространенные виды запросов. Однако существуют и другие виды запросов, которые, например, выполняются на базе предварительно созданного запроса на выборку. К ним относятся, прежде всего:

- ❑ *запросы с параметром* (интересны тем, что критерий отбора может задать сам пользователь, введя нужный параметр при вызове запроса);
- ❑ *итоговые запросы*, назначение которых отдаленно напоминает итоговые функции электронных таблиц (производят математические вычисления по заданному полю и выдают результат);
- ❑ *запросы на изменение* — позволяют автоматизировать заполнение полей таблиц;
- ❑ *перекрестные запросы*, позволяющие создавать результирующие таблицы на основе результатов расчетов, полученных при анализе группы таблиц;
- ❑ *специфические запросы SQL* — запросы к серверу базы данных, написанные на языке запросов SQL.

С некоторыми видами запросов мы познакомимся в упражнениях.

Работа с формами

С одной стороны, формы позволяют пользователям вводить данные в таблицы базы данных без непосредственного доступа к самим таблицам. С другой стороны, они позволяют выводить результаты работы запросов не в виде скупых результирующих таблиц, а в виде красивых форм. В связи с таким разделением существует два вида организации структуры форм: на основе таблицы и на основе запроса, хотя возможен и комбинированный подход, — это вопрос творчества.

Для автоматического создания простой формы, основанной на одном источнике данных, можно воспользоваться командой Создание ► Формы ► Форма, предварительно выделив в области навигации тот объект (запрос или таблицу), на основе которого будет создана форма (рис. 13.14).

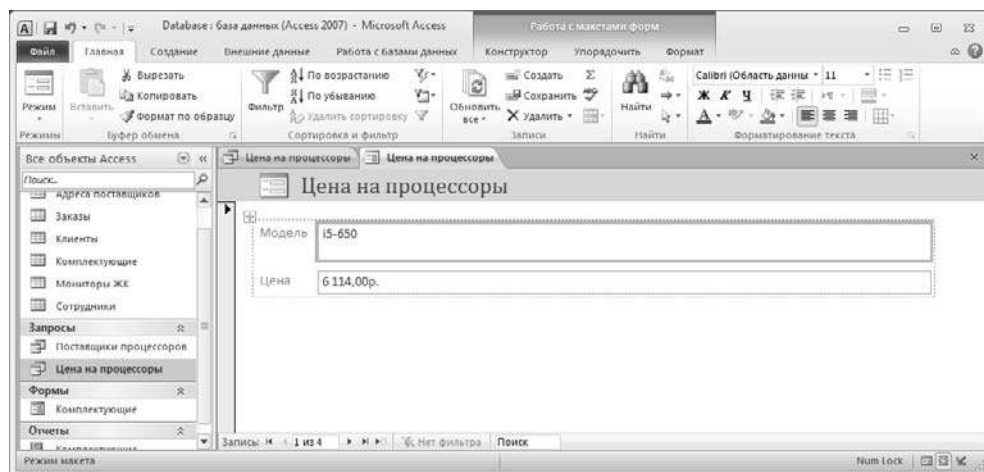


Рис. 13.14. Форма, созданная автоматически

Обратите внимание на то, что автоматически созданная форма основывается только на одном объекте. Иные средства создания форм позволяют закладывать в основу структуры формы поля нескольких таблиц или запросов. Если форма основывается только на одном объекте, она называется *простой формой*. Если форма основывается на полях из нескольких связанных таблиц, то она называется *сложной* и представляет собой композицию из нескольких форм.

Microsoft Access 2010 поддерживает различные виды форм. Так, с помощью Мастера форм (Создание ► Формы ► Мастер форм) можно создавать формы, следующих видов: «в один столбец», ленточные, табличные и выровненные. При определении свойств таких форм в их основу можно положить несколько объектов.

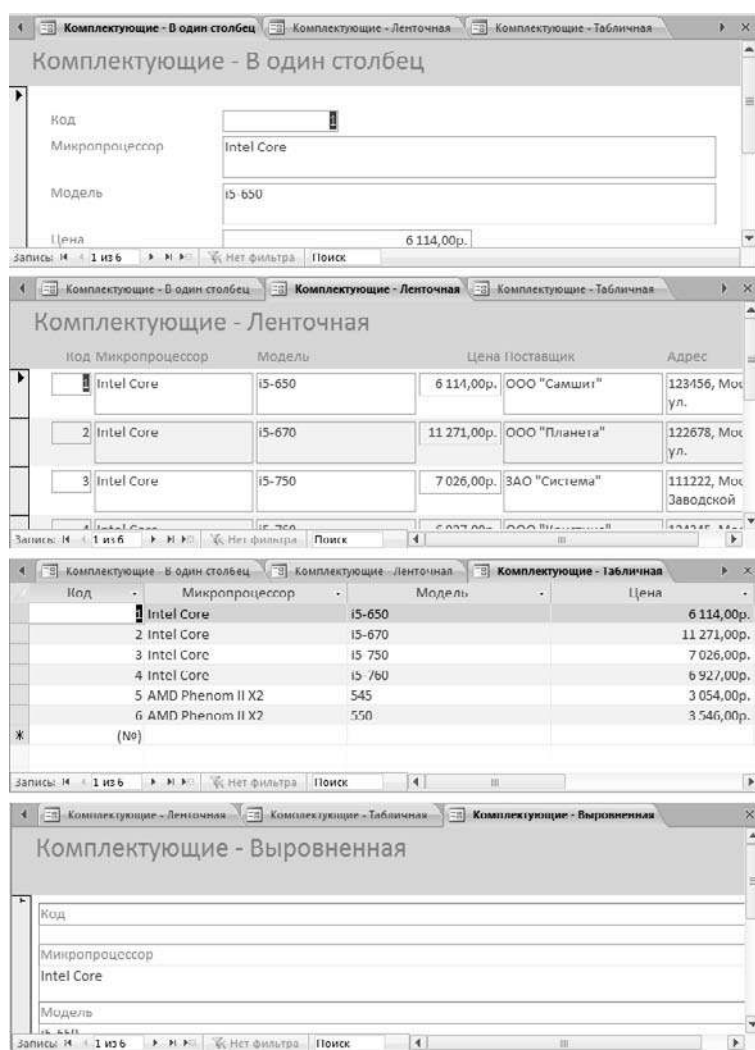


Рис. 13.15. Виды форм, ярлыки содержат указание о виде формы

Кроме того, программа поддерживает некоторые специальные виды форм. Например, это так называемые разделенные формы. Эти формы поддерживают одновременный вывод данных в двух представлениях — в виде формы и в виде таблицы. Две части разделенной формы синхронизированы друг с другом.

В некоторых случаях нужно, чтобы одна из форм открывалась в окне программы автоматически при открытии файла базы данных. Для того чтобы сделать одну из форм автозапускаемой, достаточно перейти по пути **Файл** ▶ **Параметры** ▶ **Текущая база данных**, найти в появившемся окне группу **Параметры приложений** и в списке **Форма просмотра** выбрать нужную форму. Прежде чем форма станет доступной в данном списке, она должна быть создана и сохранена.

Создание форм с помощью мастера. Автоматизированные средства предоставляет **Мастер форм** — специальное программное средство, создающее структуру формы в режиме диалога с разработчиком.

Мастер форм можно запустить командой **Создание** ▶ **Формы** ▶ **Мастер форм**.

1. На первом этапе работы **Мастера форм** выбирают таблицы и поля, которые войдут в будущую форму.
2. На втором этапе выбирается способ представления данных в том случае, если форма включает в себя поля различных объектов. Возможно представление таких данных в виде связанных или подчиненных форм.
3. На третьем этапе выбирается стиль оформления формы. Здесь, в зависимости от исходных данных формы, возможны такие варианты, как в один столбец, ленточный, табличный, выровненный.
4. На последнем этапе выполняется сохранение формы под заданным именем. Здесь же можно включить переключатель **Изменить макет формы**, который открывает только что созданную форму в режиме **Конструктора**. Этим удобно воспользоваться в учебных целях, чтобы рассмотреть структуру формы на готовом примере.

Структура формы. Как видно из рис. 13.16, форма имеет три основных раздела: область заголовка, область данных и область примечания. Линии, разделяющие разделы, перетаскиваются по вертикали с помощью мыши — это позволяет изменять размеры разделов так, как требуется.

The screenshot shows a software window titled "Комплектующие - Ленточная". Inside, the form is divided into three horizontal sections. The top section, "Заголовок формы", contains the title "Комплектующие - Ленточная" and a row of six fields: "Код", "Микропроцессор", "Модель", "Цена", "Поставщик", and "Адрес". The middle section, "Область данных", contains an identical row of six fields. The bottom section, "Примечание формы", is currently empty. The window has a standard Windows-style title bar and a scroll bar on the right.

Рис. 13.16. Структура формы, созданной мастером

Разделы заголовка и примечания имеют чисто оформительское назначение — их содержимое напрямую не связано с таблицей или запросом, на котором основана форма. Раздел данных имеет содержательное значение — в нем представлены элементы управления, с помощью которых выполняется отображение данных или их ввод. Разработчик формы может разместить здесь дополнительные элементы управления (рис. 13.17) для автоматизации ввода данных (переключатели, флажки, списки и другие, типичные для приложений *Windows*).

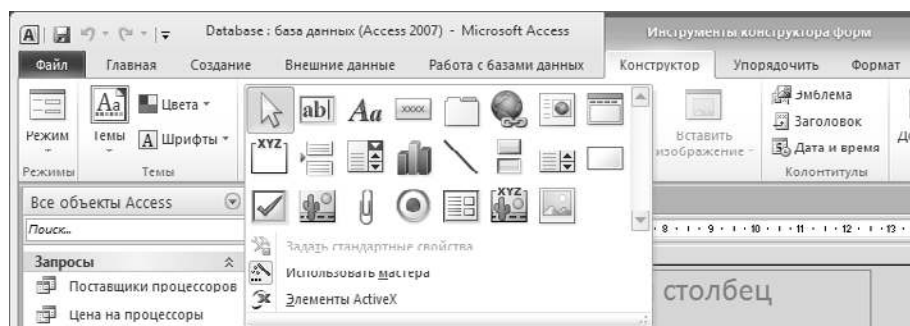


Рис. 13.17. Элементы управления

Элементы управления формы. Элементы управления, которыми может пользоваться разработчик, представлены на вкладке Конструктор ► Элементы управления. Эта вкладка принадлежит группе Инструменты конструктора форм, которая становится доступной при работе с формой в режиме конструктора.

Выбор элемента управления выполняется одним щелчком на его значке, после чего следующим щелчком в поле формы отмечается место, куда он должен быть поставлен. Вместе с элементом в поле формы вставляется его *присоединенная надпись*. По умолчанию эта надпись стандартная, например для переключателей это Переключатель1, Переключатель2 и т. д. Редактированием свойства элемента управления (доступ к свойствам открывается через контекстное меню) можно дать элементу управления более содержательную подпись.

Основными элементами оформления формы являются текстовые надписи и рисунки. Для создания в форме текстовых надписей служат два элемента управления — Надпись и Поле. В качестве надписи можно задать произвольный текст. Элемент Поле отличается тем, что в нем отображается содержимое одного из полей таблицы, на которой основана форма, то есть при переходе от записи к записи текст может меняться.

Для создания графических элементов оформления служат элементы управления Рисунок, Свободная рамка объекта и Присоединенная рамка объекта. Рисунок выбирается из графического файла и вставляется в форму. Элемент Свободная рамка объекта отличается тем, что это не обязательно рисунок — это может быть любой другой объект OLE, например мультимедийный. Элемент Присоединенная рамка объекта тоже в какой-то степени может служить для оформления формы, но его содержимое берется не из назначенного файла, а непосредственно из таблицы базы

данных (если она имеет поле объекта OLE). Естественно, что при переходе между записями содержимое этого элемента будет меняться.

Дизайн формы. В то время как таблицы базы данных глубоко скрыты от посторонних глаз, формы базы данных — это средства, с помощью которых с ней общаются люди. Поэтому к формам предъявляются повышенные требования по дизайну.

В первую очередь, все элементы управления форм должны быть аккуратно выровнены. Это обеспечивается командой Упорядочить ► Размер и порядок ► Выровнять. Если нужно равномерно распределить элементы управления по полю формы, используют средства меню Упорядочить ► Размер и порядок ► Размер или интервал.

Ручное изменение размеров и положения элементов управления тоже возможно, но редко приводит к качественным результатам. При работе вручную используют перетаскивание маркеров, которые видны вокруг элемента управления, когда он выделен. Особый статус имеет маркер левого верхнего угла. Обычно элементы управления перетаскиваются вместе с присоединенными к ним надписями. Перетаскивание с помощью этого маркера позволяет оторвать присоединенную надпись от элемента. Существенную помощь при разработке дизайна формы оказывает вспомогательная сетка. Ее отображение включают командой Упорядочить ► Размер и порядок ► Размер или интервал ► Сетка. Автоматическую привязку элементов к узлам сетки включают командой Упорядочить ► Размер и порядок ► Размер или интервал ► Сетка ► Привязать.

Управление последовательностью перехода. Пользователь, для которого, собственно, и разрабатывается форма, ожидает, что ввод данных в нее должен происходить по элементам управления слева направо и сверху вниз. Однако при проектировании сложных форм, когда в процессе дизайна элементы управления многократно перемещаются с места на место, очень легко перепутать их последовательность и создать неудобный порядок ввода данных.

Физически последовательность перехода — это порядок перехода к следующему полю по окончании работы с предыдущим. Она легко проверяется с помощью клавиши TAB. Если при последовательных нажатиях этой клавиши фокус ввода «мечется» по всей форме, значит, последовательность перехода нерациональна и ее надо править (рис. 13.18).

Для управления последовательностью перехода служит диалоговое окно Последовательность перехода (Конструктор ► Сервис ► Переходы). В нем представлен список элементов управления формы. Порядок элементов в списке соответствует текущему порядку перехода. Изменение порядка перехода выполняется перетаскиванием в два приема:

- ❑ щелчком на кнопке маркера слева от названия выделяется элемент управления (кнопка мыши отпускается);
- ❑ после повторного щелчка с перетаскиванием элемент перемещается на новое место.

Закончив разработку макета формы, ее следует закрыть и сохранить под заданным именем. После открытия формы в области навигации с ней можно работать: просматривать или редактировать данные из базовой таблицы. Проверку последовательности перехода выполняют клавишей TAB.

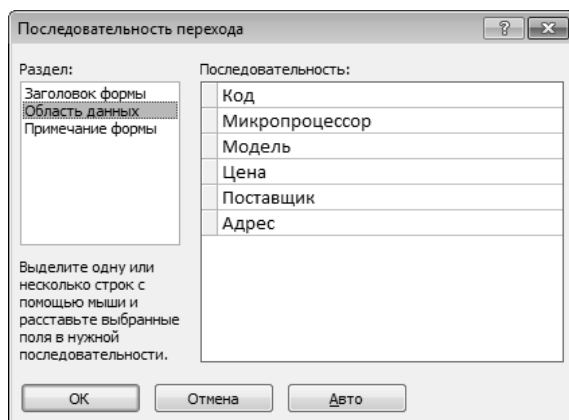


Рис. 13.18. Окно Последовательность перехода

Работа с отчетами

Отчеты во многом похожи на формы, но имеют иное функциональное назначение — они служат для форматированного вывода данных на печатающие устройства и, соответственно, при этом должны учитывать параметры принтера и параметры используемой бумаги.

Большая часть того, что было сказано о формах, относится и к отчетам. Здесь также существуют средства автоматического, автоматизированного и ручного проектирования.

Средства автоматического проектирования реализованы автоотчетами (Создание ► Отчеты ► Отчет).

Средством автоматизированного создания отчетов является Мастер отчетов. Он запускается командой Создание ► Отчеты ► Мастер отчетов. Мастер отчетов работает в шесть этапов. При его работе выполняется выбор базовых таблиц или запросов, на которых отчет базируется, выбор полей, отображаемых в отчете, выбор полей группировки, выбор полей и методов сортировки, выбор формы печатного макета и стиля оформления.

Структура готового отчета отличается от структуры формы только увеличенным количеством разделов. Кроме разделов заголовка, примечания и данных, отчет может содержать разделы верхнего и нижнего колонтитулов. Если отчет занимает более одной страницы, эти разделы необходимы для печати служебной информации, например номеров страниц. Чем больше страниц занимает отчет, тем важнее роль данных, выводимых на печать через эти разделы. Если для каких-то полей отчета применена группировка, количество разделов отчета увеличивается, поскольку оформление заголовков групп выполняется в отдельных разделах.

Редактирование структуры отчета выполняют в режиме Конструктора (режим запускается командой Главная ► Режим ► Конструктор). Приемы редактирования те же, что и для форм. Элементы управления в данном случае выполняют функции элементов оформления, поскольку печатный отчет не интерактивный объект, в отличие от электронных форм.

Размещение элементов управления выполняют с помощью элементов управления вкладки Конструктор группы ► Инструменты конструктора отчетов, которая по составу практически не отличается от аналогичной вкладки конструктора форм. Важной особенностью отчетов является наличие средства для вставки в область верхнего или нижнего колонтитула текущего номера страницы, полного количества страниц и некоторых других данных. Эти операции выполняют с помощью элементов управления, сгруппированных по адресу Конструктор ► Колонтитулы.

Практическое занятие

Упражнение 13.1. Создание базовых таблиц

Руководитель малого предприятия, выполняющего сборку персональных компьютеров из готовых компонентов, заказал разработку базы данных, основанной на двух таблицах комплектующих. Одна таблица содержит данные, которые могут отображаться для клиентов при согласовании спецификации изделия, — в ней указаны розничные цены на компоненты. Вторая таблица предназначена для анализа результатов деятельности предприятия — в ней содержатся оптовые цены на компоненты и краткая информация о поставщиках (клиенты предприятия не имеют доступа к данным этой таблицы).

1. Запустите программу Microsoft Access (Пуск ► Все программы ► Microsoft Office ► Microsoft Access 2010).
2. На панели Создать файл, в области Доступные шаблоны щелкните на значке Новая база данных, задайте имя файла базы данных (Комплектующие), убедитесь, что в качестве папки назначения базы данных указана папка Документы текущего пользователя, и нажмите на кнопку Создать. Откроется окно новой базы — Комплектующие: база данных.

При создании базы данных будет автоматически создана новая таблица. Она будет открыта в режиме таблицы. Выполните команду Работа с таблицами ► Поля ► Режимы ► Конструктор. Система попросит задать имя для таблицы. В появившемся окне Сохранение, в поле Имя таблицы введите Комплектующие — откроется бланк создания структуры таблицы.

1. Для первой таблицы введите следующие поля.

Имя поля	Тип поля
Компонент	Текстовый
Модель	Текстовый
Основной параметр	Числовой
Цена	Числовой

Обратите внимание на то, что в данном случае поле Цена задано не денежным типом, а числовым. Данные, относящиеся к денежному типу, имеют размерность, выраженную в рублях (если работа выполняется с версией Microsoft Access, локализованной в России). Но в некоторых организациях стоимость компонентов вычислительной техники выражать в этой единице измерения не

принято. Для сравнимости цен разных поставщиков обычно используют «условные единицы». В таких случаях удобно использовать поле числового типа, чтобы не перенастраивать всю СУБД.

- Щелкните на поле Цена (рис. 13.19). В нижней части бланка задайте свойство Число десятичных знаков равным 2.
- Для связи с будущей таблицей поставщиков надо задать ключевое поле. Поскольку здесь ни одно поле явно не претендует на «уникальность», используем комбинацию полей Компонент и Модель. Выделите оба поля в верхней части бланка (при нажатой клавише SHIFT). Щелчком правой кнопки мыши откройте контекстное меню и выберите в нем пункт Ключевое поле.

Рис. 13.19. Таблица Комплектующие в режиме Конструктора

- Сохраните созданную таблицу командой Сохранить контекстного меню ярлычка таблицы.
- Создайте таблицу Поставщики (Создание ► Таблицы ► Таблица), переключитесь в режим конструктора, задайте имя таблицы Поставщики. В новую таблицу должны входить следующие поля.

Имя поля	Тип поля
Компонент	Текстовый
Модель	Текстовый
Цена оптовая	Числовой
Поставщик	Текстовый
Телефон	Текстовый
Адрес	Текстовый
Примечание	Поле МЕМО

Обратите внимание на то, что поле номера телефона является текстовым, несмотря на то что обычно номера телефонов записывают цифрами. Это связано с тем, что они не имеют числового содержания. Номера телефонов не сравнивают по величине, не вычитают из одного номера другой и т. д. Это типичное текстовое поле. Ключевое поле можно не задавать — для текущей задачи оно не требуется.

6. В окне **Комплектующие: база данных** откройте по очереди созданные таблицы и наполните их экспериментальным содержанием (3–4 записи). Закончив работу, закройте таблицы и завершите работу с программой.

Итог. Мы научились создавать таблицы базы данных, задавать их структуру, выбирать типы полей и управлять их свойствами. Мы также освоили приемы наполнения таблиц конкретным содержанием.

Упражнение 13.2. Создание межтабличных связей

1. Откройте папку **Документы** и двойным щелчком по файлу **Комплектующие.accdb** откройте базу данных, созданную в предыдущем упражнении.
2. Убедитесь, что в панели навигации присутствуют таблицы **Комплектующие** и **Поставщики**.
3. Выполните команду **Работа с базами данных ► Отношения ► Схема данных**. Одновременно с открытием окна **Схема данных** открывается диалоговое окно **Добавление таблицы**, на вкладке **Таблицы** которого можно выбрать таблицы, между которыми создаются связи.
4. Щелчком на кнопке **Добавить** выберите таблицы **Комплектующие** и **Поставщики** — в окне **Схема данных** откроются списки полей этих таблиц.
5. При нажатой клавише **SHIFT** выделите в таблице **Комплектующие** два поля — **Компонент** и **Модель**.
6. Перетащите эти поля на список полей таблицы **Поставщики**. При отпускании кнопки мыши автоматически откроется диалоговое окно **Изменение связей**.
7. На правой панели окна **Изменение связей** выберите поля **Компонент** и **Модель** таблицы **Поставщики**, включаемые в связь. Не устанавливайте флажок **Обеспечение целостности данных**: в данном упражнении это не требуется, но может препятствовать постановке учебных опытов с таблицами.
8. Щелкните на кнопке **Создать** в диалоговом окне **Изменение связей**. В окне **Схема данных** рассмотрите образовавшуюся связь. Убедитесь в том, что линию связи можно выделить щелчком левой кнопки мыши, а щелчком правой кнопки мыши открывается контекстное меню, позволяющее разорвать связь или отредактировать ее.
9. Закройте окно **Схема данных**. Закройте программу **Microsoft Access**.

Итог. Мы изучили условия, необходимые для создания взаимосвязанных таблиц, и приемы их создания. Мы познакомились с основными типами связей, образующихся между таблицами, и научились редактировать параметры связи.

Упражнение 13.3. Создание запроса на выборку

В этом упражнении мы создадим запрос на выборку ЖК-мониторов, имеющих диагональ свыше 17 дюймов при цене менее 250 условных единиц. Результирующая таблица должна содержать также адрес поставщика и номер его телефона.

1. Откройте папку Документы и двойным щелчком по файлу Комплектующие. accdb откройте базу данных, созданную в первом упражнении.
2. Выполните команду Создание ► Запросы ► Конструктор запросов. Откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно Добавление таблицы.
3. В окне Добавление таблицы выберите таблицу Поставщики и щелкните на кнопке Добавить. Закройте окно Добавление таблицы.
4. В списке полей таблицы Поставщики выберите поля, включаемые в результирующую таблицу: Компонент, Модель, Цена оптовая, Поставщик, Телефон. Выбор производите двойными щелчками на именах полей.
5. Задайте условие отбора для поля Компонент. В соответствующую строку введите: ЖК-монитор. Из таблицы будут выбираться не все изделия, а только ЖК-мониторы.
6. Задайте условие отбора для поля Цена оптовая. В соответствующую строку введите: < 250. Из таблицы будут выбираться только изделия, имеющие цену менее 250 условных единиц.
7. Нам еще надо задать условие отбора по основному потребительскому параметру — диагонали монитора. Однако в таблице Поставщики такого поля нет. С другой стороны, в ней есть поле Модель, которое однозначно определяет параметры изделия. Благодаря тому, что по полю Модель у нас установлена связь с таблицей Комплектующие, мы получаем возможность ввести в запрос поле Основной параметр, взяв его из другой таблицы.
Добавьте список полей таблицы Комплектующие в верхнюю часть бланка запроса по образцу. Для этого щелкните правой кнопкой мыши в верхней области бланка и в открывшемся контекстном меню выберите пункт Добавить таблицу — откроется уже знакомое нам окно Добавление таблицы. Выберите в нем таблицу Комплектующие.
8. Двойным щелчком на поле Основной параметр в списке полей таблицы Комплектующие введите это поле в бланк запроса по образцу.
9. В строке Условие отбора столбца Основной параметр введите условие >17 (диагональ монитора более 17 дюймов).
10. Закройте бланк запроса по образцу. При закрытии запроса введите его имя — Выбор комплектующих.
11. В панели навигации откройте только что созданный запрос и рассмотрите результирующую таблицу. Ее содержательность зависит от того, что было введено в таблицы Комплектующие и Поставщики при их наполнении в упражнении 13.1. Если ни одно изделие не соответствует условию отбора и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы

и наполните их модельными данными, позволяющими проверить работу запроса.

12. По окончании исследований закройте все открытые объекты и завершите работу с программой Microsoft Access.

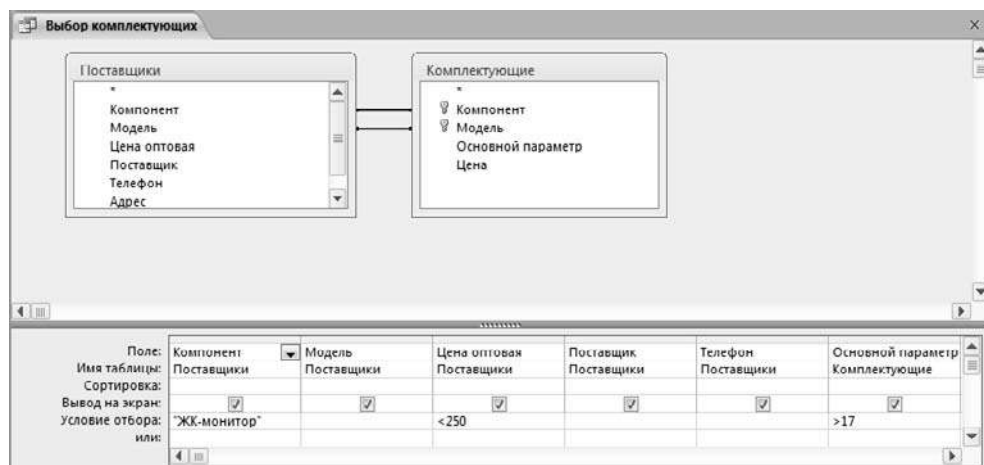


Рис. 13.20. Пример создания бланка запроса по образцу

Итог. Мы научились создавать запрос на выборку, основанный на связанных таблицах. Мы научились формировать структуру запроса путем выбора базовых таблиц и полей, а также формулировать условие отбора. Мы выяснили, какую роль играют связи между таблицами при создании запросов на выборку данных.

Упражнение 13.4. Создание запросов «с параметром»

Выше мы рассмотрели, как действует условие отбора, но должны отметить его существенный недостаток. Пользователь базы данных работает с запросами, которые ему подготовил разработчик. Если, например, разработчик предусмотрел запрос, отбирающий изделия, имеющие цену менее 250 условных единиц, то пользователь базы уже не в состоянии отобрать изделия, цена которых менее 240 условных единиц, поскольку у него нет соответствующего запроса.

Специальный тип запросов, называемый запросами «с параметром», позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этим приемом обеспечивается гибкость работы с базой.

Создадим простой запрос, позволяющий отбирать процессоры, предельную цену которых пользователь может задать сам при запуске запроса.

1. Откройте папку Документы и двойным щелчком по файлу Комплектующие.accdb откройте базу данных, созданную в первом упражнении.
2. Выполните команду Создание ► Запросы ► Конструктор запросов. Откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно Добавление таблицы.

3. В окне **Добавление таблицы** выберите таблицу **Поставщики** и щелкните на кнопке **Добавить**. Закройте окно **Добавление таблицы**.
4. В списке полей таблицы **Поставщики** выберите поля, включаемые в результирующую таблицу: **Компонент**, **Модель**, **Цена оптовая**, **Поставщик**, **Телефон**. Выбор производите двойными щелчками на именах полей.
5. В строке **Условие отбора** поля **Компонент** введите: **Процессор**.
6. Строку **Условие отбора** для поля **Цена оптовая** надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение.

Текст, обращенный к пользователю, должен быть заключен в квадратные скобки. Если бы мы хотели отобрать процессоры, цена которых больше 100 единиц, мы бы написали: **>200**. Если бы нам были нужны процессоры дешевле 80 единиц, мы бы написали **<100**. Но если мы хотим дать пользователю возможность выбора, мы должны написать: **< [Введите максимальную цену]**.

Цена оптовая
Поставщики
☒
< [Введите максимальную цену]

Рис. 13.21. Указание запрашиваемого условия

7. Закройте запрос. При закрытии сохраните его под именем **Выбор комплектующих** с параметром.
8. В окне **Комплектующие: база данных** откройте панель **Запросы** и запустите запрос **Выбор комплектующих**. На экране появится диалоговое окно **Введите значение параметра** (рис. 13.22).

Введите значение параметра

Введите максимальную цену

250

ОК

Отмена

Рис. 13.22. Пользователь вводит значение параметра для условия отбора

9. Введите какое-либо число и щелкните на кнопке **ОК**. В зависимости от того, что реально содержится в таблице **Поставщики**, по результатам запроса будет сформирована результирующая таблица.
10. Закройте все объекты базы данных. Закройте программу Microsoft Access.

Итог. Мы научились формировать запросы «с параметром» и узнали, что в основе этого вида запросов лежат запросы на выборку, у которых в поле **Условие отбора** записан заключенный в квадратные скобки текст, обращенный к пользователю.

Упражнение 13.5. Создание итогового запроса

Если полностью заполнить данными таблицу *Комплектующие*, введя параметры всех компонентов, входящих в сборочную спецификацию персонального компьютера, то можно узнать, во что обходится себестоимость комплектующих узлов. Запросы, выполняющие вычисления по всем записям для какого-либо числового поля, называются *итоговыми запросами*. В итоговом запросе может рассчитываться сумма значений или величина среднего значения по всем ячейкам поля, может выбираться максимальное или минимальное значение данных в поле, может также исполняться иная *итоговая функция*. Итоговые запросы, как и запросы на выборку, готовятся с помощью бланка запроса по образцу.

Предположим, что малое предприятие собирает компьютеры трех классов: «Элитный», «Деловой» и «Экономичный». Несмотря на то что архитектура у всех компьютеров близка, их компоненты заметно отличаются по цене и техническим параметрам. Соответственно, имеются значительные отличия в цене этих трех моделей, что важно для захвата разных секторов рынка. Наша задача — подготовить итоговый отчет, с помощью которого можно определять цену каждой из модели компьютеров и динамично ее отслеживать при изменении входящих компонентов или их поставщиков.

1. Откройте папку *Документы* и двойным щелчком по файлу *Комплектующие.accdб* откройте базу данных, созданную в первом упражнении.
2. В панели навигации выберите таблицу *Комплектующие*.
3. Командой *Главная ► Режимы ► Режим ► Конструктор* откройте таблицу в режиме проектирования — нам это необходимо для создания дополнительного поля *Класс*, в котором будут храниться данные о том, для какого класса изделий предназначены компоненты.
4. В начало структуры таблицы вставьте новое поле. Для этого выделите первое поле (*Компонент*) и нажмите клавишу *INSERT*.
5. Введите имя нового поля — *Класс* и определите его тип — *Текстовый*.
6. Закройте окно *Конструктора*. При закрытии подтвердите необходимость изменить структуру таблицы.
7. Откройте таблицу *Комплектующие* и наполните ее содержанием, введя для каждого класса данные по следующим изделиям:
 - материнская плата;
 - процессор;
 - оперативная память;
 - жесткий диск;
 - корпус;
 - дисковод CD-ROM;
 - дисковод гибких дисков;
 - видеоадаптер;
 - звуковая карта;

- клавиатура;
- мышь.

Цены на эти изделия для каждого класса проставьте произвольно. Прочие поля таблицы можно не заполнять — в формировании итогового запроса они участвовать не будут.

8. Закройте таблицу Комплектующие.
9. Выполните команду Создание ► Запросы ► Конструктор запросов. В открывшемся диалоговом окне Добавление таблицы выберите таблицу Комплектующие, на основе которой будет разрабатываться итоговый запрос. Закройте окно Добавление таблицы.
10. В бланк запроса по образцу введите следующие поля таблицы Комплектующие: Класс, Компонент, Цена.
11. Для поля Класс включите сортировку по возрастанию. Включите также сортировку по полю Цена, но на этот раз — по убыванию.
12. Выполните команду Работа с запросами ► Конструктор ► Показать или скрыть ► Итоги. Эта команда необходима для создания в нижней части бланка строки Групповая операция. Именно на ее базе и создаются итоговые вычисления. Все поля, отобранные для запроса, получают в этой строке значение Группировка.
13. Для поля, по которому производится группировка записей (в нашем случае — Класс), оставьте в строке Групповые операции значение Группировка. Для остальных полей щелкните в этой строке — появится кнопка раскрывающегося списка, из которого можно выбрать итоговую функцию для расчета значений в данном поле.
14. Для поля Цена выберите итоговую функцию Sum для определения стоимости изделия как суммы стоимостей комплектующих.
15. Для поля Компонент выберите итоговую функцию Count, определяющую общее количество записей, вошедших в группу. В нашем случае это количество узлов, из которых собран компьютер.
16. Закройте бланк запроса по образцу и дайте ему имя: Расчет стоимости изделия. Запустите запрос и убедитесь, что он правильно работает.
17. Закройте все объекты базы данных. Завершите работу с программой Microsoft Access.

Итог. Мы научились создавать так называемые *итоговые запросы*. Это запросы, производящие вычисления по всем значениям одного поля. Мы научились также выбирать используемую итоговую функцию.

Глава четырнадцатая

Приемы и методы работы со сжатыми данными

Как хранение, так и передача информации обходятся участникам информационного процесса недешево. Зная стоимость носителя и его емкость (Мбайт, Гбайт), нетрудно подсчитать, во что обходится хранение единицы информации, а зная пропускную способность канала связи (Мбит/с) и стоимость его аренды, можно определить затраты на передачу единицы информации. Полученные результаты обычно составляют вполне значимые величины как для корпоративных, так и для индивидуальных пользователей. В связи с этим регулярно возникает необходимость сжимать данные перед тем, как размещать их в архивах или передавать по каналам связи. Соответственно, существует и обратная необходимость восстановления данных из предварительно уплотненных архивов.

14.1. Теоретические основы сжатия данных

Характерной особенностью большинства «классических» типов данных, с которыми традиционно работают люди, является определенная избыточность. Степень избыточности зависит от типа данных. Например, у видеоданных степень избыточности обычно в несколько раз больше, чем у графических данных, а степень избыточности графических данных в несколько раз больше, чем текстовых. Кроме того, степень избыточности данных зависит от принятой системы кодирования. Так, например, можно сказать, что кодирование текстовой информации средствами русского языка (с использованием русской азбуки) дает в среднем избыточность на 20–30% больше, чем кодирование адекватной информации средствами английского языка.

Для человека избыточность информации нередко связана с представлением о ее качестве, поскольку избыточность, как правило, улучшает восприятие, особенно в неблагоприятных условиях (просмотр телепередач при наличии помех, восстановление поврежденного графического материала, чтение текстов в условиях недостаточной освещенности и т. п.).

При обработке информации избыточность также играет важную роль. Так, например, при преобразовании или селекции информации избыточность используют для повышения ее качества (репрезентативности, актуальности, адекватности и т. п.). Однако когда речь заходит не об обработке, а о хранении готовых документов или их передаче, то избыточность можно уменьшить, что дает эффект сжатия данных.

Если методы сжатия информации применяют к готовым документам, то нередко термин *сжатие данных* подменяют термином *архивация данных*, а программные средства, выполняющие эти операции, называют *архиваторами*.

Объекты сжатия

В зависимости от того, в каком объекте размещены данные, подвергаемые сжатию, различают:

- ☐ уплотнение (архивацию) файлов;
- ☐ уплотнение (архивацию) папок;
- ☐ уплотнение дисков.

Уплотнение файлов применяют для уменьшения их размеров при подготовке к передаче по каналам электронных сетей или к транспортировке на внешнем носителе малой емкости, например на гибком диске.

Уплотнение папок используют как средство архивации данных перед длительным хранением, в частности при резервном копировании.

Уплотнение дисков служит целям повышения эффективности использования их рабочего пространства и, как правило, применяется к дискам, имеющим недостаточную емкость.

Обратимость сжатия

Несмотря на изобилие алгоритмов сжатия данных, теоретически есть только три способа уменьшения их избыточности. Это либо изменение содержания данных, либо изменение их структуры, либо и то и другое вместе.

Если при сжатии данных происходит изменение их содержания, метод сжатия необратим и при восстановлении данных из сжатого файла не происходит полного восстановления исходной последовательности. Такие методы называют также *методами сжатия с регулируемой потерей информации*. Они применимы только для тех типов данных, для которых формальная утрата части содержания не приводит к значительному снижению потребительских свойств. В первую очередь это относится к мультимедийным данным: видеорядам, музыкальным записям, звукозаписям и рисункам. Методы сжатия с потерей информации обычно обеспечивают гораздо более высокую степень сжатия, чем обратимые методы, но их нельзя применять к текстовым документам, базам данных и тем более к программному коду. Характерными форматами сжатия с потерей информации являются:

- ☐ .JPG для графических данных;
- ☐ .MPG для видеоданных;
- ☐ .MP3 для звуковых данных.

Если при сжатии данных происходит только изменение их структуры, то метод сжатия обратим. Из результирующего кода можно восстановить исходный массив путем применения обратного метода. Обратимые методы применяют для сжатия любых типов данных. Характерными форматами сжатия без потери информации являются:

- ☐ .GIF, .TIF, .PCX и многие другие для графических данных;
- ☐ .AVI для видеоданных;
- ☐ .ZIP, .ARJ, .RAR, .7Z, .LZH, .LH, .CAB и многие другие для любых типов данных.

Алгоритмы обратимых методов

При исследовании методов сжатия данных следует иметь в виду существование следующих доказанных теорем.

1. Для любой последовательности данных существует теоретический предел сжатия, который не может быть превышен без потери части информации.
2. Для любого алгоритма сжатия можно указать такую последовательность данных, для которой он обеспечит лучшую степень сжатия, чем другие методы.
3. Для любого алгоритма сжатия можно указать такую последовательность данных, для которой данный алгоритм вообще не позволит получить сжатия.

Таким образом, обсуждая различные методы сжатия, следует иметь в виду, что наивысшую эффективность они демонстрируют для данных разных типов и разных объемов.

Существует достаточно много обратимых методов сжатия данных, однако в их основе лежит сравнительно небольшое количество теоретических алгоритмов, представленных в табл. 14.1.

Таблица 14.1. Свойства алгоритмов сжатия

Алгоритм	Выходная структура	Сфера применения	Примечание
RLE (Run-Length Encoding)	Список (вектор данных)	Графические данные	Эффективность алгоритма не зависит от объема данных
KWE (Keyword Encoding)	Таблица данных (словарь)	Текстовые данные	Эффективен для массивов большого объема
Алгоритм Хафмана	Иерархическая структура (дерево кодировки)	Любые данные	Эффективен для массивов большого объема

Алгоритм RLE

В основу алгоритмов RLE положен принцип выявления повторяющихся последовательностей данных и замены их простой структурой, в которой указывается код данных и коэффициент повтора.

Например, для последовательности: 0; 0; 0; 127; 127; 0; 255; 255; 255; 255 (всего 10 байтов) образуется следующий вектор:

Значение	Коэффициент повтора
0	3
127	2
0	1
255	4

При записи в строку он имеет вид:

0; 3; 127; 2; 0; 1; 255; 4 (всего 8 байтов).

В данном примере коэффициент сжатия равен $8/10$ (экономия объема составляет 20%).

Программные реализации алгоритмов RLE отличаются простотой, высокой скоростью работы, но в среднем обеспечивают недостаточное сжатие. Наилучшими объектами для данного алгоритма являются графические файлы, в которых большие одноцветные участки изображения кодируются длинными последовательностями одинаковых байтов. Этот метод также может давать заметный выигрыш на некоторых типах файлов баз данных, имеющих таблицы с фиксированной длиной полей. Для текстовых данных методы RLE, как правило, не эффективны.

Алгоритм KWE

В основу алгоритмов кодирования по ключевым словам (*Keyword Encoding*) положено кодирование лексических единиц исходного документа группами байтов фиксированной длины. Примером лексической единицы может служить слово (последовательность символов, справа и слева ограниченная пробелами или символами конца абзаца). Результат кодирования сводится в таблицу, которая прикладывается к результирующему коду и представляет собой словарь. Обычно для англоязычных текстов принято использовать двухбайтную кодировку слов. Образующиеся при этом пары байтов называют *токенами*.

Эффективность данного метода существенно зависит от длины документа, поскольку из-за необходимости прикладывать к архиву словарь длина кратких документов не только не уменьшается, но даже возрастает.

Данный алгоритм наиболее эффективен для англоязычных текстовых документов и файлов баз данных. Для русскоязычных документов, отличающихся увеличенной длиной слов и большим количеством приставок, суффиксов и окончаний, не всегда удается ограничиться двухбайтными токенами, и эффективность метода заметно снижается.

Алгоритм Хаффмана

В основе этого алгоритма лежит кодирование не байтами, а битовыми группами.

- ❑ Перед началом кодирования производится частотный анализ кода документа и выявляется частота повтора каждого из встречающихся символов.
- ❑ Чем чаще встречается тот или иной символ, тем меньшим количеством битов он кодируется (соответственно, чем реже встречается символ, тем длиннее его кодовая битовая последовательность).
- ❑ Образующаяся в результате кодирования иерархическая структура прикладывается к сжатому документу в качестве таблицы соответствия.

Пример кодирования символов русского алфавита представлен на рис. 14.1.

Как видно из схемы, представленной на рис. 14.1, используя 16 бит, можно закодировать до 256 различных символов. Однако ничто не мешает использовать и последовательности длиной до 20 бит — тогда можно закодировать до 1024 лексических единиц (это могут быть не символы, а группы символов, слоги и даже слова).

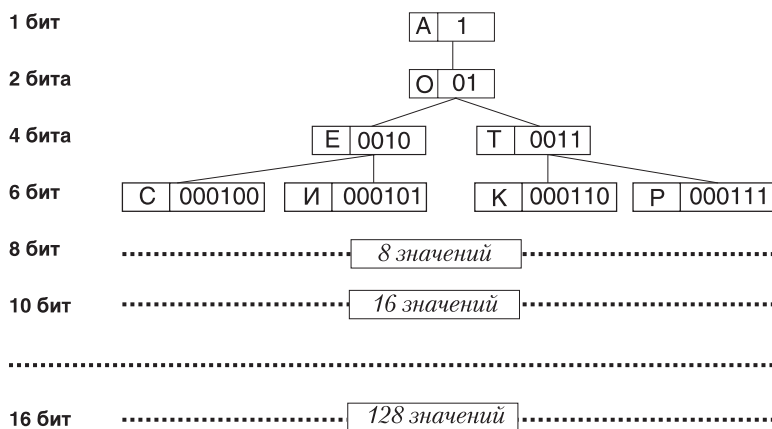


Рис. 14.1. Пример побуквенного кодирования русского алфавита по алгоритму Хаффмана

В связи с тем, что к сжатому архиву необходимо прикладывать таблицу соответствия, на файлах малых размеров алгоритм Хаффмана малоэффективен. Практика также показывает, что его эффективность зависит и от заданной предельной длины кода (размера словаря). В среднем, наиболее эффективными оказываются архивы с размером словаря от 512 до 1024 единиц (длина кода до 18–20 бит).

Синтетические алгоритмы

Рассмотренные выше алгоритмы в «чистом виде» на практике не применяют из-за того, что эффективность каждого из них сильно зависит от начальных условий. В связи с этим современные средства архивации данных используют более сложные алгоритмы, основанные на комбинации нескольких теоретических методов. Общим принципом в работе таких «синтетических» алгоритмов является предварительный просмотр и анализ исходных данных для индивидуальной настройки алгоритма на особенности обрабатываемого материала.

14.2. Программные средства сжатия данных

«Классическими» форматами сжатия данных, широко используемыми в повседневной работе с компьютером, являются форматы .ZIP, .RAR и .ARJ. Программные средства, предназначенные для создания и обслуживания архивов, выполненных в данных форматах, приведены в табл. 14.2.

Таблица 14.2. Средства архивации файлов для Windows

Формат сжатия	Средство архивации	Средство разархивирования
.ZIP	WinZip, сжатые ZIP-папки, WinRAR	
.RAR	WinRAR	
.ARJ	WinArj	WinArj, WinRAR

Стоит отметить, что в связи с широким распространением нескольких форматов сжатия многие программные средства для сжатия данных начинают приобретать универсальный характер, позволяя упаковывать и распаковывать сжатые архивы разных типов. Программные средства для Windows обычно имеют один «предпочтительный» тип архива, но также справляются с распаковкой данных при работе с архивами других типов.

Наиболее распространен формат .ZIP, который является стандартом де-факто для архивов, распространяемых через Интернет. Немаловажную роль в этом играет открытость этого формата. Этот формат является полностью открытым — его использование не требует никаких лицензионных отчислений.

Операционная система Windows 7 позволяет рассматривать ZIP-архивы как сжатые папки. Это предполагает полностью «прозрачную» работу с такими архивами — все файловые операции можно выполнять в сжатой папке так же, как в обычной. Однако специализированные средства работы с архивами обеспечивают более широкий набор функций.

При использовании файловой системы NTFS возможно внутреннее сжатие при записи на диск для экономии пространства. С точки зрения всех файловых операций такие файлы не рассматриваются как упакованные. При их копировании на носитель с другой файловой системой они автоматически распаковываются.

Базовые требования к диспетчерам архивов

Современные программные средства для создания и обслуживания архивов отличаются большим объемом функциональных возможностей, многие из которых выходят далеко за рамки простого сжатия данных и эффективно дополняют стандартные средства операционной системы. В этом смысле современные средства архивации данных называют *диспетчерами архивов*.

К базовым функциям, которые выполняют большинство современных диспетчеров архивов, относятся:

- ☐ извлечение файлов из архивов;
- ☐ создание новых архивов;
- ☐ добавление файлов в имеющийся архив;
- ☐ создание самораспаковывающихся архивов;
- ☐ создание распределенных архивов на носителях малой емкости;
- ☐ тестирование целостности структуры архивов;
- ☐ полное или частичное восстановление поврежденных архивов;
- ☐ защита архивов от просмотра и несанкционированной модификации.

Самораспаковывающиеся архивы. В тех случаях, когда архивация производится для передачи документа потребителю, следует предусмотреть наличие у него программного средства, необходимого для извлечения исходных данных из уплотненного архива. Если таких средств у потребителя нет или нет оснований предполагать их наличие, создают самораспаковывающиеся архивы. Самораспаковывающийся архив готовится на базе обычного архива путем присоединения

к нему небольшого программного модуля. Сам архив получает расширение имени .EXE, характерное для исполнимых файлов. Потребитель сможет выполнить его запуск как программы, после чего распаковка архива произойдет на его компьютере автоматически.

Распределенные архивы. В тех случаях, когда предполагается передача большого архива на носителях малой емкости, например на гибких дисках, или передача большого объема данных через Интернет, возможно распределение одного архива в виде малых фрагментов на нескольких носителях.

Современные диспетчеры архивов способны выполнить предварительное разбиение архива на фрагменты заданного размера на жестком диске. Впоследствии их можно перенести на внешние носители путем копирования. Все файлы распределенного архива получают разные имена, и поэтому их последующее упорядочение не вызывает проблем.

Оптимальный режим работы с распределенными архивами следующий:

- ☐ создание набора файлов распределенного архива в папке на жестком диске;
- ☐ копирование файлов распределенного архива на отдельные сменные носители (обычно гибкие диски) или передача по электронной почте, выгрузка на FTP-сервер;
- ☐ перенос (перевозка) сменных носителей в место назначения;
- ☐ копирование файлов распределенного архива со сменных носителей в одну папку на конечном жестком диске или, при электронном способе передачи информации, загрузка их в эту папку;
- ☐ распаковка распределенного архива.

Гибкие диски являются крайне ненадежными носителями, поэтому архивы, сформированные на жестком диске, должны храниться до получения подтверждения от потребителя о том, что распределенный архив поступил к нему в неповрежденном виде и прошел распаковку. Правилом «хорошего тона» считается создание двух копий при передаче материалов на гибких дисках.

Защита архивов. В большинстве случаев защиту архивов выполняют с помощью пароля, который запрашивается при попытке просмотреть, распаковать или изменить архив. Теоретически, защита с помощью пароля считается неудовлетворительной и не рекомендуется для особо важной информации. В то же время необходимо отметить, что основные программные средства, используемые для восстановления утраченного пароля (или взлома закрытой информации, что, по сути, то же самое), используют методы прямого перебора. Работу этих средств можно существенно затруднить и замедлить, если расширить область перебора. Пароли на базе только букв английского алфавита и цифр действительно снимаются очень быстро. Однако даже незначительное увеличение числа используемых символов за счет знаков препинания многократно увеличивает криптостойкость защиты, а использование также и символов русского алфавита может полностью опровергнуть попытки снять пароль путем перебора, сделав сроки работы неприемлемыми.

Дополнительные требования к диспетчерам архивов

К дополнительным функциям диспетчеров архивов относятся сервисные функции, делающие работу более удобной. Они часто реализуются внешним подключением дополнительных служебных программ и обеспечивают:

- ☐ просмотр файлов различных форматов без извлечения их из архива;
- ☐ поиск файлов и данных внутри архивов;
- ☐ установку программ из архивов без предварительной распаковки;
- ☐ проверку отсутствия компьютерных вирусов в архиве до его распаковки;
- ☐ криптографическую защиту архивной информации;
- ☐ декодирование сообщений электронной почты;
- ☐ «прозрачное» уплотнение исполнимых файлов .EXE и .DLL;
- ☐ создание самораспаковывающихся многотомных архивов;
- ☐ выбор или настройку коэффициента сжатия информации.

Практическое занятие

Упражнение 14.1. Просмотр архивного файла в формате .ZIP

1. Запустите диспетчер архивов WinZip 14.5 командой Пуск ► Все программы ► WinZip ► WinZip 14.5.
2. Найдите архивный файл в папке, указанной преподавателем.
3. Дайте команду Главное меню ► Открыть.
4. В диалоговом окне Открыть файл WinZip воспользуйтесь областью переходов, разыщите заданную папку, выделите значок файла в формате .ZIP и щелкните на командной кнопке Открыть.

В окне программы WinZip откроется список файлов, входящих в архив (рис. 14.2).

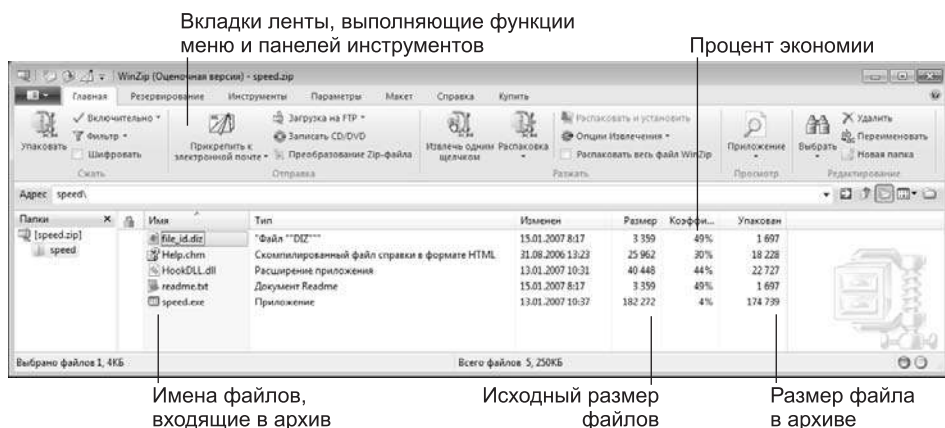


Рис. 14.2. Просмотр содержимого архива с помощью программы WinZip

5. Выделите значок файла, который желаете просмотреть, например `readme.txt`. Дважды щелкните на этом значке, и файл откроется в окне ассоциированной программы. По умолчанию в операционной системе *Windows 7* с файлами типа `.TXT` ассоциирована программа Блокнот.
6. Если нужно просмотреть файл в другой программе, выделите его значок и воспользуйтесь командой Главная ► Просмотр ► Приложение. По этой команде откроется меню, в котором можно самостоятельно указать, какую программу следует использовать в качестве средства просмотра для файлов данного типа.
7. Выделите для просмотра файл с иным расширением имени, например `file_id.diz`. Этот тип файлов не имеет ассоциированной программы просмотра. Поэтому программа выведет при попытке открыть его стандартное окно операционной системы, сообщающее о том, что файл открыть не удалось, и предлагающее пользователю либо найти соответствие файла и программы для его открытия в Интернете, либо сделать это вручную.

Итог. Мы научились просматривать файлы, находящиеся внутри архива, без распаковки архива и, соответственно, без извлечения файлов. Это важно, поскольку далеко не все архивы, поступающие на компьютер, особенно из Интернета, стоит распаковывать. Предварительный просмотр позволяет убедиться в том, что архив содержит действительно необходимые данные, а также получить предварительную информацию о том, как лучше распорядиться его содержимым.

Упражнение 14.2. Извлечение файлов из ZIP-архива

1. Предварительно определите папку, в которую будут помещены файлы, извлекаемые из архива. Если никаких предпочтений нет, создайте временную папку, например папку `\Документы\Zip_Test`.
2. Запустите диспетчер архивов *WinZip* и откройте ZIP-архив, размещенный в папке, указанной преподавателем.
3. Выделите файлы, которые желаете извлечь. При групповом выделении пользуйтесь левой кнопкой мыши совместно с клавишами **CTRL** и **SHIFT**.
4. Дайте команду Главная ► Разжать ► Распаковка — откроется диалоговое окно Распаковка, представленное на рис. 14.3.
5. С помощью стандартных средств окна найдите нужную папку.
6. Запустите процесс извлечения файлов щелчком на командной кнопке Распаковка.
7. По окончании процесса будет открыта целевая папка, в которую было произведено извлечение данных из архива. Убедитесь в том, что файлы, извлеченные из архива, действительно поступили в заданную папку.
8. Удалите экспериментальную папку `\Zip_Test`.

Итог. Мы научились извлекать файлы из архива в заданную папку. При этом у нас есть следующие возможности выбора:

- ☐ какие файлы извлекать, а какие — нет;
- ☐ в какую папку производить запись.

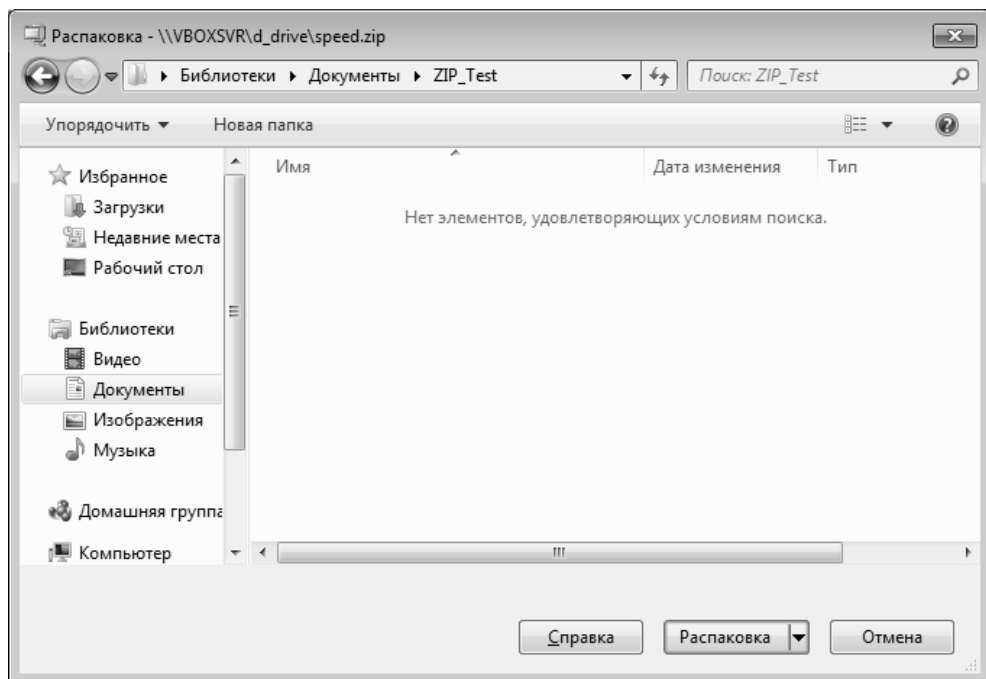


Рис. 14.3. Управление извлечением файлов из архива

Упражнение 14.3. Создание ZIP-архива

1. Создайте папку для экспериментов, например \Документы\ Zip_Test.
2. Методом копирования наполните эту папку произвольными файлами, например, взяв их из стандартной папки \Windows\Media (файлы в этой папке имеют «длинные имена», а нам важно убедиться в том, что после архивации они остаются неповрежденными).
3. Найдите папку, которую вы хотите упаковать, с помощью Проводника Windows.
4. Вызовите контекстное меню папки и выберите в нем команду WinZip ► Добавить в Zip-файл. При необходимости, для добавления в архив отдельных файлов, а не всей папки, можно выделить их с помощью мыши и клавиш CTRL и SHIFT на клавиатуре.
5. Архиватор WinZip будет запущен автоматически, откроется окно Добавить (рис. 14.4).
6. В раскрывающемся списке Действие выберите пункт Добавить (и заменить) файлы.
7. В группе Метод сжатия выберите .Zip: устаревший метод сжатия (максимальная совместимость). Установка этой опции гарантирует открытие созданного вами архива практически на любом компьютере, оснащенном архиватором WinZip как последней, так и более старых версий.

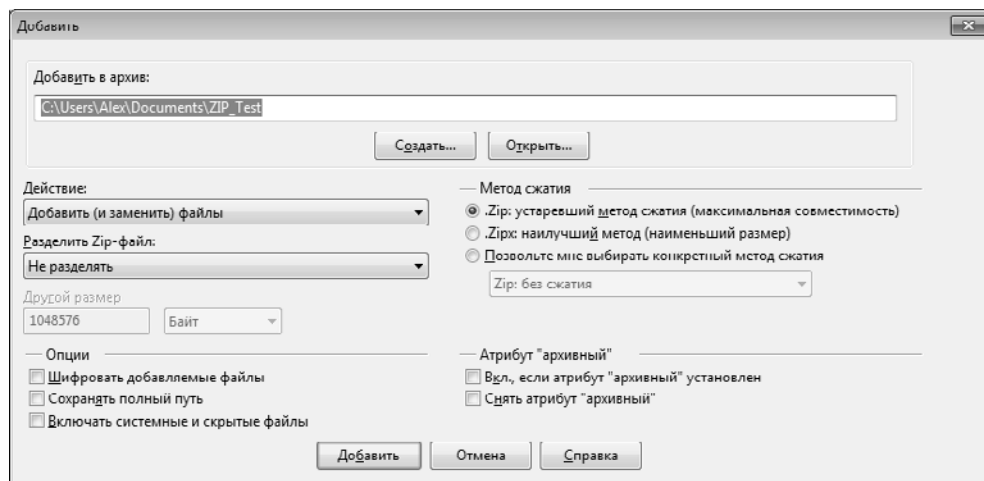


Рис. 14.4. Управление добавлением файлов в архив

8. В группе элементов управления Опции убедитесь в том, что флажок Включать системные и скрытые файлы установлен.
9. По умолчанию архив будет иметь имя папки, для которой вызывалась команда создания архива. Если вам нужен архив с другим именем, нажмите на кнопку Создать и укажите имя архива.
10. Щелкните на командной кнопке Добавить в архив — начнется процесс создания архива.
11. С помощью Проводника откройте папку \Документы. Разыщите созданный файл архива. Установите его размер — по строке состояния или переключившись в режим Таблица.
12. Переключитесь в программу WinZip. Откройте в ней созданный архив и убедитесь в том, что «длинные имена» файлов сохранились.
13. Закройте программу WinZip.

Итог. Мы научились создавать архивы сжатых файлов. При этом мы имеем возможность выбрать, какие файлы включаются в архив и в какой папке создается результирующий файл. Мы также можем выбрать степень сжатия, указать режим обновления файлов для тех случаев, когда в архиве уже имеются файлы, одноименные добавляемым, и задать режим упаковки папок с сохранением структуры вложенных папок.

Данные настройки выполняются в специальном диалоговом окне Добавить. Они особенно важны в тех случаях, когда диспетчер архивов используется с целью организации резервного копирования данных.

Создавать архивы можно не только используя контекстное меню файла или папки, как было рассмотрено выше, но и пользуясь основной программой.

Упражнение 14.4. Защита ZIP-архива

1. Запустите программу WinZip.
2. Для защиты создаваемого архива паролем установите флажок Главная ► Сжать ► Шифровать.
3. Выполните команду Главная ► Сжать ► Упаковать.
4. В появившемся окне Упаковать выберите файлы и папки, которые вы хотите поместить в архив, и нажмите на кнопку Упаковать.
5. В появившемся окне Шифровать (рис. 14.5), введите пароль и подтверждение пароля.

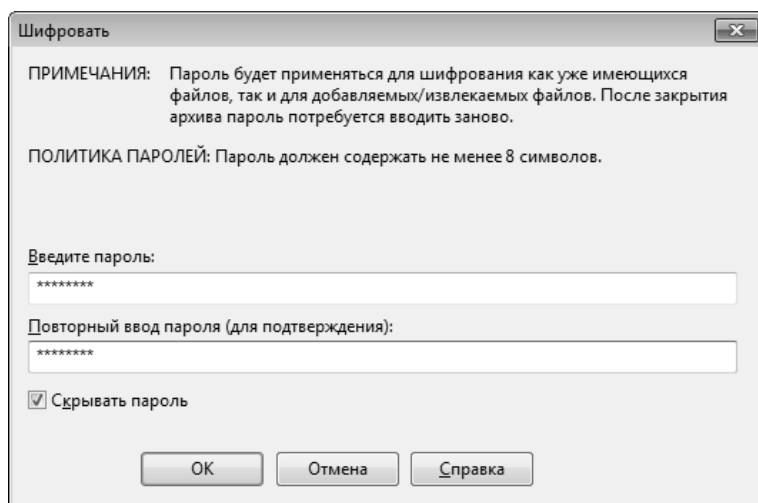


Рис. 14.5. Задание пароля для защиты архива

6. Обратите внимание на флажок Скрывать пароль. Если он установлен, пароль при вводе не будет отображаться на экране, а его символы будут заменены подстановочным символом «*». Это мера защиты пароля от посторонних. Однако в данном случае пользователь не может быть уверен в том, что он набрал пароль правильно. Поэтому при установленном флажке Скрывать пароль система запрашивает повторный (контрольный) ввод пароля.
7. Введите пароль, например слово test, используя кроме алфавитных символов произвольные знаки препинания и меняя регистр символов, например так: t,e/s:T.
8. Щелкните на кнопке ОК. Откроется окно Новый файл WinZip. Введите в поле Имя файла (Media в нашем случае) имя для файла архива, который вы хотите создать.
9. Щелкните на командной кнопке ОК — начнется процесс создания защищенного архива Media.zip.
10. Откройте созданный архив командой Меню приложения ► Открыть.

11. Сделайте двойной щелчок по любому файлу архива. Процесс извлечения и открытия файла не запустится, вместо него будет выдано окно для ввода пароля **Расшифровать**.
12. Убедитесь в том, что ввод неправильного пароля не позволяет просмотреть или извлечь файлы из архива.
13. Убедитесь в том, что правильный ввод пароля действительно запускает процесс.
14. Закройте программу WinZip.

Итог. Мы научились создавать защищенные архивы. Защита выполняется одновременно с созданием самого архива. Надежность защиты во многом определяется размером символьного набора, использованного при задании пароля.

Упражнение 14.5. Создание самораспаковывающегося ZIP-архива

1. Запустите программу WinZip.
2. Дайте команду Меню программы ► Открыть. Откройте ранее созданный архив Media.zip.
3. Дайте команду Инструменты ► Сборка ► Самораспаковывающийся EXE — откроется диалоговое окно WinZip Self-Extractor (Генератор самораспаковывающегося архива), представленное на рис. 14.6.
4. Самораспаковывающийся архив создается на базе ранее созданного обычного архива. Убедитесь в том, что в поле Создать самораспаковывающийся архив из ... правильно записан адрес исходного ZIP-файла. Если это не так, воспользуйтесь кнопкой Обзор для поиска нужного файла.
5. В группе Поддержка разбивки включите переключатель Без разбивки.
6. Щелкните на кнопке ОК.
7. Поскольку исходный архив имеет парольную защиту, то перед началом преобразования появится предупреждающее сообщение. Закройте его щелчком на кнопке ОК.
8. По окончании преобразования появится диалоговое окно с предупреждением о том, что самораспаковывающийся архив следует испытать. Подтвердите начало испытания щелчком на кнопке Да.
9. В очередном окне выберите папку, в которую произойдет проверочная распаковка архива. По умолчанию предлагается папка \Temp. Если она не устраивает, воспользуйтесь командной кнопкой Обзор, выберите нужную папку и щелкните на кнопке Распаковать.
10. Поскольку архивный файл защищен, его распаковка не начнется, пока не будет введен правильный пароль.
11. По завершении работы закройте текущие диалоговые окна.

Итог. Мы научились создавать самораспаковывающиеся архивы на базе предварительно подготовленных обычных архивов. Мы убедились в том, что если исходный архив был защищен паролем, то эта защита сохраняется и в самораспаковывающемся архиве.

Упражнение 14.6. Создание самораспаковывающегося распределенного архива

1. Запустите программу WinZip.
2. Дайте команду Меню программы ► Открыть. Разыщите и откройте ранее созданный архив Media.zip.
3. Дайте команду Инструменты ► Сборка ► Самораспаковывающийся EXE — откроется диалоговое окно WinZip Self-Extractor (Генератор самораспаковывающегося архива), представленное выше (см. рис. 14.6).

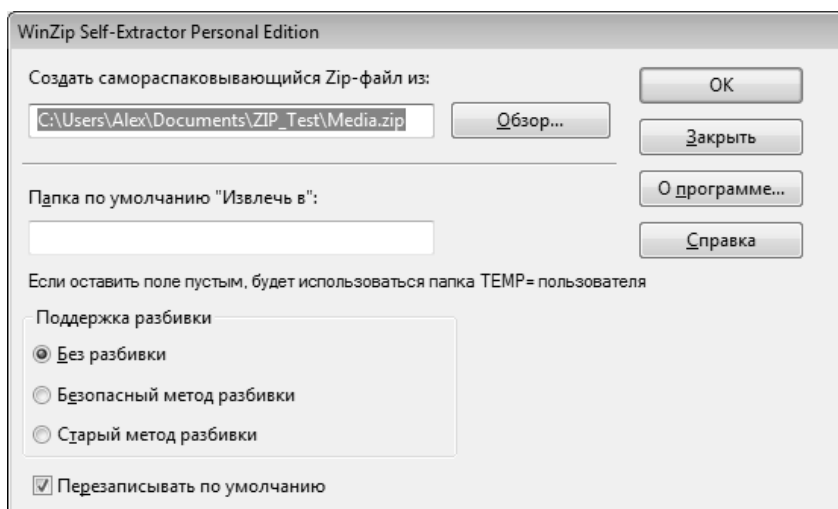


Рис. 14.6. Создание самораспаковывающегося архива

4. В группе элементов управления Поддержка разбивки включите переключатель Безопасный метод разбивки или Старый метод разбивки.

Защищенный метод создает на первом гибком диске два файла: исполнимый файл, выполняющий автоматическую распаковку, и первый том распределенного архива. На последующих дисках создается продолжение распределенного архива. Такой подход повышает уровень безопасности, поскольку даже в том случае, когда исполнимый файл поврежден, например, компьютерным вирусом, информация не пропадает и остается в архивном файле. После применения антивирусных средств для «лечения» зараженного исполнимого файла также не происходит нарушений в структуре архива, поскольку исполнимый файл автономен. Этот метод применяют для передачи архивных материалов на гибких дисках.

Обычный метод обеспечивался прошлыми версиями программы WinZip. В этом случае на первом диске создается один файл, содержащий и архив, и исполняемый модуль, а все остальные диски также содержат файлы с расширением .EXE, несмотря на отсутствие в них исполняемого кода. Последствия случайного запуска файла, не являющегося первым файлом архива, могут оказаться непредсказуемыми.

5. Вставьте в дисковод гибкий диск и щелкните на кнопке ОК — начнется процесс создания первого тома распределенного архива.
6. Следуйте указаниям программы.

Итог. Мы научились создавать самораспаковывающиеся распределенные архивы. Такой метод архивации нашел особенно широкое применение при распространении программных продуктов.

Упражнение 14.7. Наблюдение интеграционных свойств программы WinZip

В этом упражнении на примере диспетчера архивов WinZip мы ознакомимся с тем, как выполняется интеграция служебных и прикладных программ с операционной системой Windows 7.

Под интеграцией программного обеспечения понимают возможность совместной работы нескольких различных программ в рамках единой системы управления. Так, например, уже известным нам системным средством интеграции является концепция внедрения и связывания объектов и основанный на ней буфер обмена Windows. Здесь же мы познакомимся с другим приемом интеграции, в основе которого лежит изменение свойств программы Проводник и связанного с ней контекстного меню объектов. Выше мы уже пользовались запуском процесса создания архива из контекстного меню группы выделенных файлов. Это один из примеров интеграции.

1. Запустите программу Проводник (пункт Открыть проводник контекстного меню кнопки Пуск).
2. Создайте папку для экспериментов C:\Temp (если эта папка уже существует, создайте в ней папку C:\Temp\Test).
3. Скопируйте в созданную папку несколько произвольных файлов, взяв их, например, из папки \Windows\Media.
4. Откройте созданную папку и выделите один из помещенных в нее файлов.
5. Щелкните правой кнопкой мыши на значке выделенного файла — откроется контекстное меню, представленное на рис. 14.7. Обратите внимание на то, что в нем имеются два пункта для создания архива (создание архива с произвольным именем и с именем, соответствующим текущему файлу), а также некоторые дополнительные команды. Появление этих пунктов связано с наличием в компьютерной системе диспетчера архивов и интеграции WinZip с Проводником Windows.
6. Дайте команду Добавить в Zip-файл — произойдет автоматический запуск диспетчера архивов WinZip и открытие диалогового окна Добавить.
7. В поле Добавить в архив автоматически заносится адрес текущей папки. Сюда остается вписать только название файла создаваемого архива, например C:\Temp\Test\media. Того же эффекта можно добиться, нажав на кнопку Создать и указав в появившемся окне имя создаваемого архива. Если надо открыть другую папку, это можно сделать с помощью командной кнопки Открыть. Проверив настройку прочих элементов управления, запустите процесс архивации щелчком на командной кнопке Добавить.

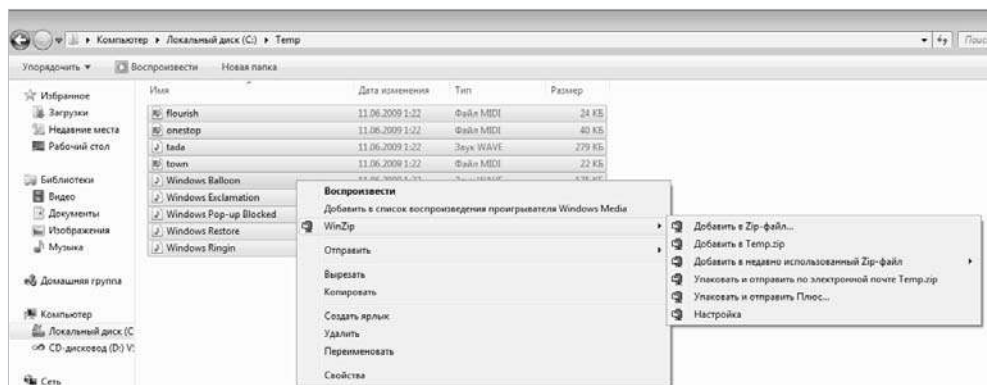


Рис. 14.7. Контекстное меню с интегрированными командами работы с архивами

8. Перейдите в окно программы Проводник и убедитесь в том, что в экспериментальной папке появился архивный файл Media.zip.
9. Щелкните на значке архивного файла правой кнопкой мыши и изучите новые команды контекстного меню, позволяющие выполнить операции с архивным файлом (рис. 14.8).
10. Дайте команду Создать самораспаковывающийся архив (.Exe). В открывшемся диалоговом окне щелкните на командной кнопке ОК и в последующих диалоговых окнах откажитесь от проверки созданного архива. Закройте открытые окна программы WinZip и в программе Проводник убедитесь в том, что в экспериментальной папке появился исполнимый файл (.EXE).

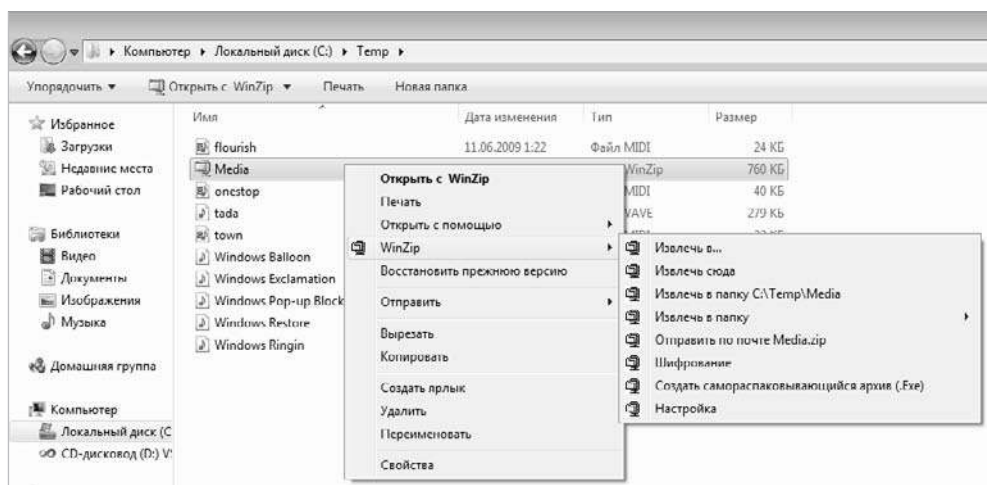


Рис. 14.8. Контекстное меню архивного файла

11. В программе Проводник выполните перетаскивание значка любого файла (или группы файлов) на значок созданного ZIP-архива. При отпускании кнопки

мыши в конце перетаскивания происходит автоматическое добавление новых файлов в архив. Если содержимое правой панели Проводника открыто в режиме Таблица, после каждого перетаскивания можно наблюдать увеличение размера файла архива.

Итог. Мы наблюдали факт изменения свойств программы Проводник за счет того, что в компьютерной системе установлена программа WinZip. Это изменение свойств выразилось, во-первых, в изменении содержания контекстного меню, связанного с объектами Windows, и, во вторых, в том, что работает прием наполнения архива методом перетаскивания. Таким образом, мы можем говорить о том, что установка программы WinZip расширила функциональные возможности системной программы Проводник.

Исследовательская работа

Задание 14.1. Исследование свойств форматов сжатия графических данных

1. Подготовьте для экспериментов папку C:\Temp\Pictures.
2. Откройте графический редактор Paint (Пуск ► Все программы ► Стандартные ► Paint) и загрузите в него заранее подготовленный многоцветный рисунок, например файл Background.bmp, который следует найти в папке Windows, используя стандартные средства поиска файлов.
3. Определите размер рисунка в пикселах (Paint ► Свойства). Оцените теоретический размер рисунка в 24-разрядной палитре (3 байта на точку) по формуле:

$$S = M \cdot N \cdot 3,$$

где S — размер файла с рисунком (байт); M — ширина рисунка (точек); N — высота рисунка (точек).

Запишите результат: $S =$ _____.

4. Сохраните рисунок в папку C:\Temp\Pictures, выбрав имя файла Test и назначив тип файла: 24-разрядный рисунок.
5. Повторно сохраните рисунок, выбрав то же имя Test, но назначив тип файла GIF. При сохранении произойдет потеря определенной части графической информации.
6. Восстановите рисунок, загрузив его из ранее сохраненного файла Test.bmp, и вновь сохраните его под тем же именем, но выбрав в качестве типа файла формат JPEG.
7. Запустите программу Проводник. Откройте папку C:\Temp\Pictures в режиме Таблица. Определите размеры файлов Test.bmp, Test.gif и Test.jpg. Определите коэффициент сжатия файлов (R), взяв отношения размеров файлов к теоретической величине, полученной расчетным путем в п. 3. Результаты занесите в таблицу.

Формат файла	Размер файла, Кбайт	Степень сжатия, %
24-разрядный .BMP		
.GIF		
.JPG		

8. Ответьте на вопросы.

- ☐ Что можно сказать о степени сжатия данных в формате BMP?
- ☐ Что можно сказать о степени сжатия данных в формате JPEG?

В графическом редакторе Paint дайте команду создания нового документа **Файл ▶ Создать**. Убедитесь в том, что полотно имеет размер 640×480 . Если это не так, измените его размер командой **Главная ▶ Изображение ▶ Изменить размер** или перетаскивайте маркеры изменения размера, которые можно найти в правой и нижней части созданного рисунка, до тех пор, пока в строке состояния редактора не отобразится нужный размер изображения.

9. В качестве инструмента выберите Кисть. Задайте максимальный размер кисти. Поочередно используя 8–10 разных красок, грубо закрасьте полотно.
10. Сохраните рисунок под именем **Test_1** в формате **24-разрядный рисунок**.
11. Сохраните рисунок под именем **Test_1** в формате **GIF**.
12. Восстановите рисунок из файла **Test_1.bmp**.
13. Сохраните рисунок под именем **Test_1** в формате **JPEG**.
14. С помощью программы Проводник определите размеры сохраненных файлов и заполните таблицу.

Формат файла	Размер файла, Кбайт	Степень сжатия, %
24-разрядный .BMP		
.GIF		
.JPG		

15. Уточните ранее сделанный вывод о степени сжатия данных в формате JPEG, учитывая тот факт, что в первом случае изображение было многоцветным, а во втором случае оно имело небольшое количество цветовых оттенков.
16. Ответьте на вопросы.
 - ☐ Какой формат графических данных из рассмотренных здесь наилучшим образом подходит для передачи цветного фотографического материала по каналам электронных сетей?
 - ☐ Какой формат графических данных целесообразно использовать для передачи черно-белого фотографического материала по каналам электронных сетей?
 - ☐ Какой формат наиболее благоприятен для передачи рисунков, имеющих малое количество цветовых оттенков (до 256)?

- Какой формат графических данных не годится для передачи информации по каналам электронных сетей, но удобен для хранения изображений, предназначенных для дальнейшей обработки?
17. В программе Проводник удалите экспериментальную папку C:\Temp\Pictures.

Задание 14.2. Исследование алгоритмов сжатия программы WinZip

1. Подготовьте для экспериментов две папки C:\Temp\Input и C:\Temp\Output.
2. Наполните экспериментальную папку C:\Temp\Input произвольным материалом в объеме нескольких десятков мегабайт, например, скопировав в нее папку \Windows\Help.
3. Запустите диспетчер архивов WinZip.
4. Установите параметр Параметры ► Метод сжатия ► Zip в значение Без сжатия.
5. Дайте команду Главная ► Сжать ► Упаковать, в появившемся окне выберите папку C:\Temp\Input. Нажмите на кнопку Упаковать, задайте имя и расположение для архивного файла (имя файла Test.zip, папка C:\Temp\Output).
6. Зафиксируйте время начала архивации по секундомеру и щелкните на командной кнопке Создать.
7. Зафиксируйте время конца архивации и определите продолжительность процесса.
8. Измените режим сжатия на Сверхбыстрая, создайте файл Test_2.zip (Меню приложения ► Новый), поместите в новый архив те же файлы и замерьте продолжительность процесса.
9. Создайте файл Test_3 в режиме сжатия Максимальная и замерьте продолжительность.

Степень сжатия	Исходный размер, Мбайт	Результирующий размер, Мбайт	Время, с	Степень сжатия, %	Эффективность метода
Без сжатия					
Сверхбыстрая					
Максимальная					

Определите степень сжатия по формуле:

$$R = \frac{S_r}{S_i} \times 100\%,$$

где S_r — размер результирующего файла, Мбайт; S_i — размер исходного файла, Мбайт; R — степень сжатия.

Эффективность метода оцените по абсолютной величине приращения степени сжатия к приращению времени сжатия:

$$E_i = \frac{|R_i - R_{j-1}|}{|t_i - t_{i-1}|}.$$

10. Сделайте вывод о наиболее эффективном методе сжатия по критерию соотношения степени сжатия и расхода времени на операцию.
11. В программе Проводник удалите экспериментальные папки C:\Temp\Input и C:\Temp\Output.

Глава пятнадцатая

Введение в компьютерную графику

15.1. Основы представления графических данных

Виды компьютерной графики

Представление данных на мониторе компьютера в графическом виде впервые было реализовано в середине 50-х годов для больших ЭВМ, применявшихся в научных и военных исследованиях. С тех пор графический способ отображения данных стал неотъемлемой принадлежностью подавляющего числа компьютерных систем, в особенности персональных. Графический интерфейс пользователя сегодня является стандартом «де-факто» для программного обеспечения разных классов, начиная с операционных систем.

Существует специальная область информатики, изучающая методы и средства создания и обработки изображений с помощью программно-аппаратных вычислительных комплексов, — *компьютерная графика*. Она охватывает все виды и формы представления изображений, доступных для восприятия человеком либо на экране монитора, либо в виде копии на внешнем носителе (бумага, киноплёнка, ткань и пр.). Без компьютерной графики невозможно представить себе не только компьютерный, но и обычный, вполне материальный мир. Визуализация данных находит применение в самых разных сферах человеческой деятельности. Для примера назовем медицину (компьютерная томография), научные исследования (визуализация строения вещества, векторных полей и других данных), моделирование тканей и одежды, опытно-конструкторские разработки.

В зависимости от способа формирования изображений компьютерную графику принято подразделять на *растровую, векторную и фрактальную*.

Отдельным предметом считается *трехмерная (3D) графика*, изучающая приемы и методы построения объемных моделей объектов в виртуальном пространстве. Как правило, в ней сочетаются векторный и растровый способы формирования изображений.

Особенности цветового охвата характеризуют такие понятия, как *черно-белая и цветная графика*. На специализацию в отдельных областях указывают названия некоторых разделов: *инженерная графика, научная графика, веб-графика, компьютерная полиграфия* и пр.

На стыке компьютерных, телевизионных и кинотехнологий зародилась и стремительно развивается сравнительно новая область *компьютерной графики и анимации*.

Заметное место в компьютерной графике отведено развлечениям. Появилось даже такое понятие, как механизм графического представления данных (*Graphics Engine*) в играх. Рынок игровых программ имеет оборот в десятки миллиардов долларов и часто инициализирует очередной этап совершенствования графики и анимации.

Хотя компьютерная графика служит всего лишь инструментом, ее структура и методы основаны на передовых достижениях фундаментальных и прикладных

наук: математики, физики, химии, биологии, статистики, программирования и множества других. Это замечание справедливо как для программных, так и для аппаратных средств создания и обработки изображений на компьютере. Поэтому компьютерная графика является одной из наиболее бурно развивающихся отраслей информатики и во многих случаях выступает «локомотивом», тянущим за собой всю компьютерную индустрию.

Растровая графика

Для растровых изображений, состоящих из точек, особую важность имеет понятие *разрешения*, выражающее количество точек, приходящихся на единицу длины. При этом следует различать:

- ☐ разрешение оригинала;
- ☐ разрешение экранного изображения;
- ☐ разрешение печатного изображения.

Разрешение оригинала. Разрешение оригинала измеряется в *точках на дюйм* (*dots per inch — dpi*) и зависит от требований к качеству изображения и размеру файла, способу оцифровки или методу создания исходной иллюстрации, избранному формату файла и другим параметрам. В общем случае действует правило: чем выше требования к качеству, тем выше должно быть разрешение оригинала.

Разрешение экранного изображения. Для экранных копий изображения элементарную точку раstra принято называть *пикселом*. Размер пиксела варьируется в зависимости от выбранного *экранного разрешения* (из диапазона стандартных значений), *разрешения оригинала* и масштаба отображения.

Мониторы для обработки изображений с диагональю 19–24 дюйма (профессионального класса), как правило, обеспечивают стандартные экранные разрешения 640×480 , 800×600 , 1024×768 , 1280×1024 , 1600×1200 , 1600×1280 , 1920×1440 , 1920×1600 , 2048×1536 точек. Расстояние между соседними точками люминофора у качественного монитора составляет 0,22–0,25 мм.

Для экранной копии достаточно разрешения 72 dpi, для распечатки на цветном или лазерном принтере 150–200 dpi, для вывода на фотоэкспонирующем устройстве 200–300 dpi. Установлено эмпирическое правило, что при распечатке величина разрешения оригинала должна быть в 1,5 раза больше, чем *линиатура раstra* устройства вывода. В случае если твердая копия будет увеличена по сравнению с оригиналом, эти величины следует умножить на коэффициент масштабирования.



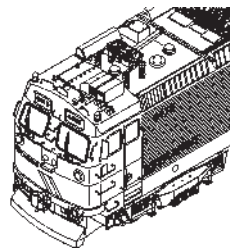
Растровая графика



Векторная графика



Трехмерная графика



Инженерная графика

Разрешение печатного изображения и понятие линиатуры. Размер точки растрового изображения как на твердой копии (бумага, пленка и т. д.), так и на экране зависит от примененного метода и параметров *растрирования* оригинала. При растрировании на оригинал как бы накладывается сетка линий, ячейки которой образуют *элемент растра*. Частота сетки растра измеряется числом *линий на дюйм* (*lines per inch — lpi*) и называется *линнатурой*.

Размер точки растра рассчитывается для каждого элемента и зависит от интенсивности тона в данной ячейке. Чем больше интенсивность, тем плотнее заполняется элемент растра. То есть если в ячейку попал абсолютно черный цвет, размер точки растра совпадет с размером элемента растра. В этом случае говорят о 100%-й заполняемости. Для абсолютно белого цвета значение заполняемости составит 0%. На практике заполняемость элемента на отпечатке обычно составляет от 3 до 98%. При этом все точки растра имеют одинаковую оптическую плотность, в идеале приближающуюся к абсолютно черному цвету. Иллюзия более темного тона создается за счет увеличения размеров точек и, как следствие, сокращения пробельного поля между ними при одинаковом расстоянии между центрами элементов растра (рис. 15.1). Такой метод называют растрированием с *амплитудной модуляцией (АМ)*.

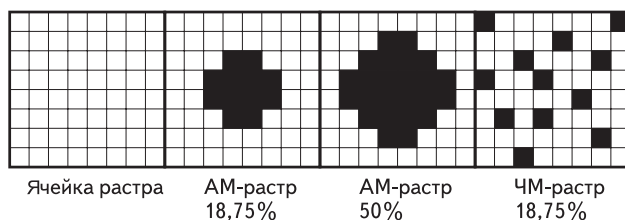


Рис. 15.1. Примеры амплитудной и частотной модуляции растра

Существует и метод растрирования с *частотной модуляцией (ЧМ)*, когда интенсивность тона регулируется изменением расстояния между соседними точками одинакового размера. Таким образом, при частотно-модулированном растрировании в ячейках растра с разной интенсивностью тона находится разное число точек (см. рис. 15.1). Изображения, растрированные ЧМ-методом, выглядят более качественно, так как размер точек минимален и, во всяком случае, существенно меньше, чем средний размер точки при АМ-растрировании. Еще более повышает качество изображения разновидность ЧМ-метода, называемая *стохастическим растрированием*. В этом случае рассчитывается число точек, необходимое для отображения требуемой интенсивности тона в ячейке растра. Затем эти точки располагаются внутри ячейки на расстояниях, вычисленных квазислучайным методом (на самом деле используется специальный математический алгоритм). То есть регулярная структура растра внутри ячейки, как и на изображении в целом, вообще отсутствует (рис. 15.2). Поэтому при стохастическом ЧМ-растрировании теряет смысл понятие линиатуры растра, имеет значение лишь разрешающая способность устройства вывода. Такой способ требует больших затрат вычислительных ресурсов и высокой точности полиграфического оборудования; он применяется в основном для художественных работ, при печати с числом красок, превышающим четыре.

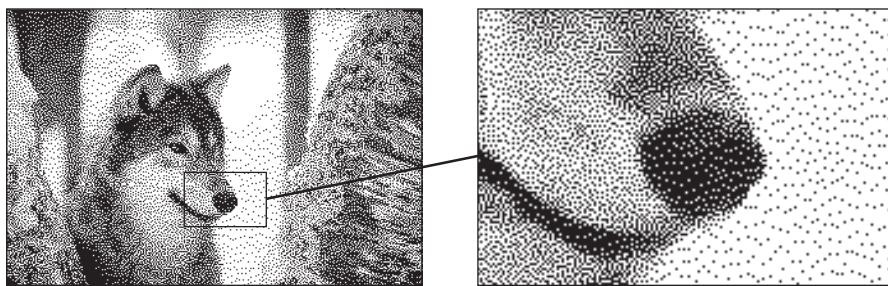


Рис. 15.2. Пример использования стохастического растра

Интенсивность тона (так называемую *светлоту*) принято подразделять на 256 уровней. Большое число градаций не воспринимается зрением человека и является избыточным. Меньшее число ухудшает восприятие изображения (минимально допустимым для качественной полутоновой иллюстрации принято значение 150 уровней). Нетрудно подсчитать, что для воспроизведения 256 уровней тона достаточно иметь размер ячейки растра $256 \times 16 \times 16$ точек.

Между разрешением оригинала, частотой растра и градацией уровней существует зависимость, описываемая формулой:

$$N = \left(\frac{dpi}{lpi} \right)^2 + 1 \quad lpi = \frac{dpi}{\sqrt{N-1}}$$

где N — число градаций уровней тона (оттенков), dpi — разрешение устройства вывода (отображения), lpi — линиятура растра. Единица в формуле соответствует абсолютно белому цвету, когда ячейка растра вообще не заполнена.

При выводе копии изображения на принтере или полиграфическом оборудовании линиятуру растра выбирают, исходя из компромисса между требуемым качеством, возможностями аппаратуры и параметрами печатных материалов. Для лазерных принтеров рекомендуемая линиятура составляет 65–100 lpi, для газетного производства — 65–85 lpi, для книжно-журнального — 85–133 lpi, для художественных и рекламных работ — 133–300 lpi.

При печати изображений с наложением растров друг на друга, например многоцветных, каждый последующий растр поворачивается на определенный угол. Традиционными для цветной печати считаются углы поворота: 105 градусов для голубой печатной формы, 75 градусов для пурпурной, 90 градусов для желтой и 45 градусов для черной. При этом ячейка растра становится косоугольной, и для воспроизведения 256 градаций тона с линиятурой 150 lpi уже недостаточно разрешения $16 \times 150 = 2400$ dpi. Поэтому для фотоэкспонирующих устройств профессионального класса принято минимальное стандартное разрешение 2540 dpi, обеспечивающее качественное растривание при разных углах поворота растра. Таким образом, коэффициент, учитывающий поправку на угол поворота растра, для цветных изображений составляет 1,06.

Динамический диапазон. Качество воспроизведения тоновых изображений принято оценивать *динамическим диапазоном* (D). Это *оптическая плотность*,

численно равная десятичному логарифму величины, обратной *коэффициенту пропускания* $\times$ (для оригиналов, рассматриваемых «на просвет», например слайдов) или *коэффициенту отражения* $\times$ (для прочих оригиналов, например полиграфических отпечатков):

$$D = \lg \frac{1}{\tau}; \quad D = \lg \frac{1}{\rho}; \quad \rho = \frac{F_{\tau}}{F_0}; \quad \tau = \frac{F_{\tau}}{F_0},$$

где F_0 — падающий световой поток, F_{ρ} — отраженный световой поток, F_{τ} — пропущенный световой поток.

Для оптических сред, пропускающих свет, динамический диапазон лежит в пределах от 0 до 4. Для поверхностей, отражающих свет, значение динамического диапазона составляет от 0 до 2. Чем выше динамический диапазон, тем большее число полутонов присутствует в изображении и тем лучше качество его восприятия.

Связь между параметрами изображения и размером файла. Средствами растровой графики принято иллюстрировать работы, требующие высокой точности в передаче цветов и полутонов. Однако размеры файлов растровых иллюстраций стремительно растут с увеличением разрешения. Фотоснимок, предназначенный для домашнего прочтения (стандартный размер 10×15 см, оцифрованный с разрешением 200–300 dpi, цветовое разрешение 24 бита), занимает в формате *TIFF* с включенным режимом сжатия около 4 Мбайт. Оцифрованный с высоким разрешением слайд занимает 45–50 Мбайт. Цветоделенное цветное изображение формата A4 занимает 120–150 Мбайт.

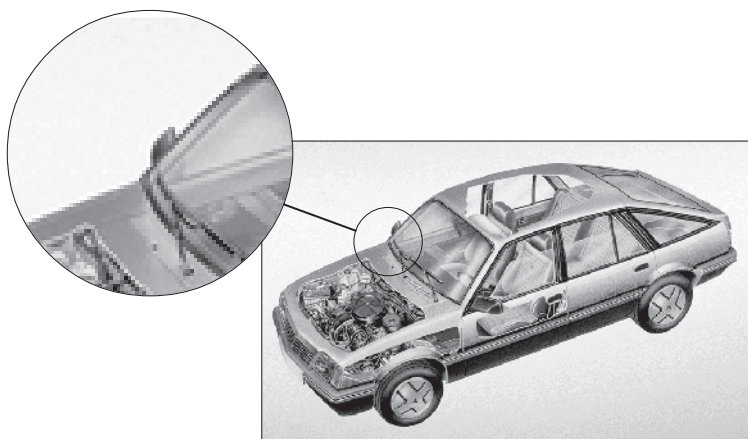


Рис. 15.3. Эффект пикселизации при масштабировании растрового изображения

Масштабирование растровых изображений. Одним из недостатков растровой графики является так называемая *пикселизация* изображений при их увеличении (если не приняты специальные меры). Раз в оригинале присутствует определенное количество точек, то при большем масштабе увеличивается и их размер, становятся заметны элементы раstra, что искажает саму иллюстрацию (рис. 15.3). Для проти-

действия пикселизации принято заранее оцифровывать оригинал с разрешением, достаточным для качественной визуализации при масштабировании. Другой прием состоит в применении стохастического растра, позволяющего уменьшить эффект пикселизации в определенных пределах. Наконец, при масштабировании используют метод интерполяции, когда увеличение размера иллюстрации происходит не за счет масштабирования точек, а путем добавления необходимого числа промежуточных точек.

Векторная графика

Если в растровой графике базовым элементом изображения является точка, то в векторной графике — *линия*. Линия описывается математически как единый объект, и потому объем данных для отображения объекта средствами векторной графики существенно меньше, чем в растровой графике.

Линия — элементарный *объект* векторной графики. Как и любой объект, линия обладает свойствами: формой (прямая, кривая), толщиной, цветом, начертанием (сплошная, пунктирная). Замкнутые линии приобретают свойство *заполнения*. Охватываемое ими пространство может быть заполнено другими объектами (*текстуры, карты*) или выбранным цветом.

Простейшая незамкнутая линия ограничена двумя точками, именуемыми *узлами*. Узлы также имеют свойства, параметры которых влияют на форму конца линии и характер сопряжения с другими объектами.

Все прочие объекты векторной графики составляются из линий. Например, куб можно составить из шести связанных прямоугольников, каждый из которых, в свою очередь, образован четырьмя связанными линиями. Возможно представить куб и как двенадцать связанных линий, образующих ребра.

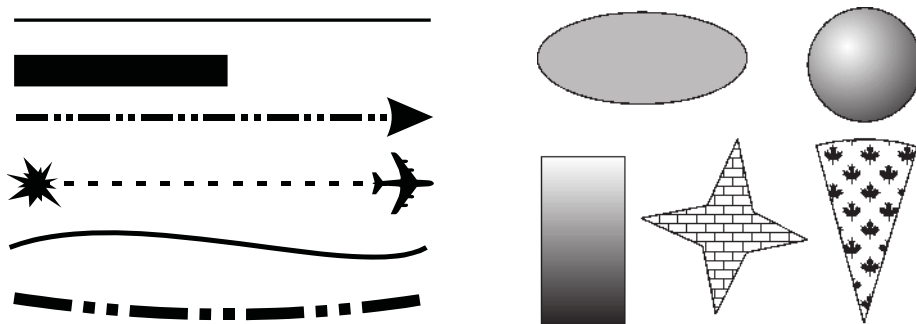


Рис. 15.4. Объекты векторной графики

Математические основы векторной графики

Рассмотрим подробнее способы представления различных объектов в векторной графике.

Точка. Этот объект на плоскости представляется двумя числами (x, y) , указывающими его положение относительно начала координат.

Прямая линия. Ей соответствует уравнение $y = kx + b$. Указав параметры k и b , всегда можно отобразить бесконечную прямую линию в известной системе координат, то есть для задания прямой достаточно двух параметров.

Отрезок прямой. Он отличается тем, что требует для описания еще двух параметров — например, координат x_1 и x_2 начала и конца отрезка.

Кривая второго порядка. К этому классу кривых относятся параболы, гиперболы, эллипсы, окружности, то есть все линии, уравнения которых содержат степени не выше второй. Кривая второго порядка не имеет *точек перегиба*. Прямые линии являются всего лишь частным случаем кривых второго порядка. Формула кривой второго порядка в общем виде может выглядеть, например, так:

$$x^2 = a_1 y^2 = a_2 xy = a_3 x = a_4 y = a_5 = 0.$$

Таким образом, для описания бесконечной кривой второго порядка достаточно пяти параметров. Если требуется построить отрезок кривой, понадобятся еще два параметра.

Кривая третьего порядка. Отличие этих кривых от кривых второго порядка состоит в возможном наличии точки перегиба. Например, график функции $y = x^3$ имеет точку перегиба в начале координат (рис. 15.5). Именно эта особенность позволяет сделать кривые третьего порядка основой отображения природных объектов в векторной графике. Например, линии изгиба человеческого тела весьма близки к кривым третьего порядка. Все кривые второго порядка, как и прямые, являются частными случаями кривых третьего порядка.

В общем случае уравнение кривой третьего порядка можно записать так:

$$x^3 = a_1 y^3 = a_2 x^2 y = a_3 xy^2 = a_4 x^2 = a_5 y^2 = a_6 xy = a_7 x = a_8 y = a_9 = 0.$$

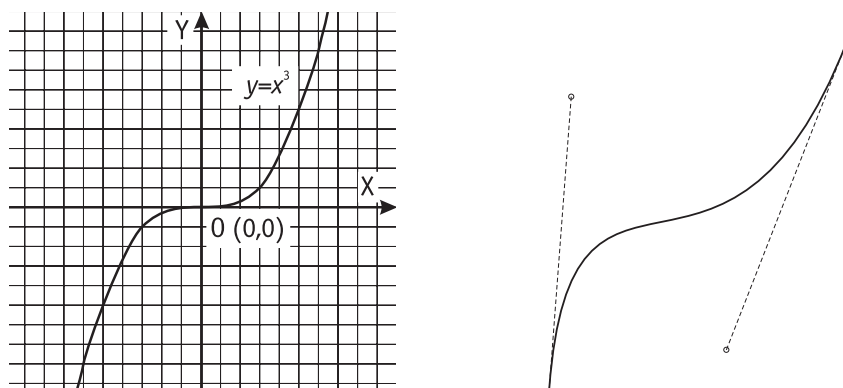


Рис. 15.5. Кривая третьего порядка (слева) и кривая Безье (справа)

Таким образом, кривая третьего порядка описывается девятью параметрами. Описание ее отрезка потребует на два параметра больше.

Кривые Безье. Это особый, упрощенный вид кривых третьего порядка (см. рис. 15.5). Метод построения кривой Безье основан на использовании пары касательных, проведенных к отрезку линии в ее окончаниях. Отрезки кривых Безье

описываются восемью параметрами, поэтому работать с ними удобнее. На форму линии влияет угол наклона касательной и длина ее отрезка. Таким образом, касательные играют роль виртуальных «рычагов», с помощью которых управляют кривой.

Фрактальная графика

Фрактальная графика, как и векторная, основана на математических вычислениях. Однако базовым элементом фрактальной графики является сама математическая формула, то есть никаких объектов в памяти компьютера не хранится, и изображение строится исключительно по уравнениям. Таким способом строят как простейшие регулярные структуры, так и сложные иллюстрации, имитирующие природные ландшафты и трехмерные объекты (рис. 15.6).

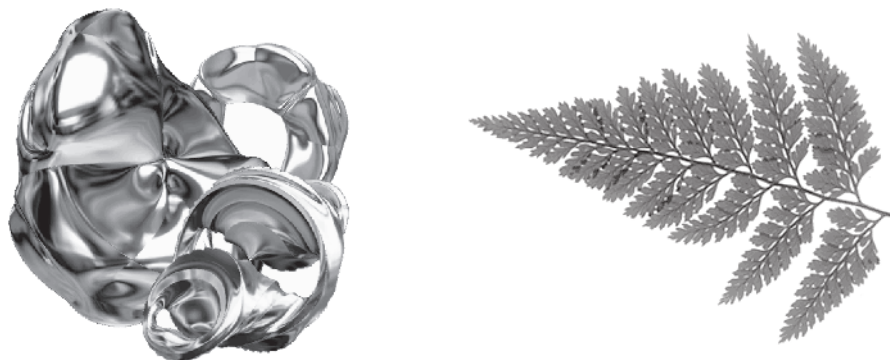


Рис. 15.6. Примеры фрактальных объектов

Основные понятия трехмерной графики

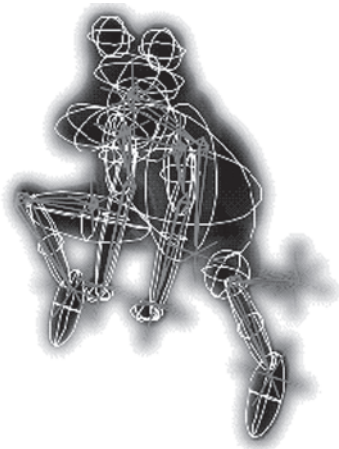
Трехмерная графика нашла широкое применение в таких областях, как научные расчеты, инженерное проектирование, компьютерное моделирование физических объектов. В качестве примера рассмотрим наиболее сложный вариант трехмерного моделирования — создание подвижного изображения реального физического тела.

В упрощенном виде для пространственного моделирования объекта требуется:

- ☐ спроектировать и создать виртуальный каркас («скелет») объекта, наиболее полно соответствующий его реальной форме;
- ☐ спроектировать и создать виртуальные материалы, по физическим свойствам визуализации похожие на реальные;
- ☐ присвоить материалы различным частям поверхности объекта (на профессиональном жаргоне — «спроектировать текстуры на объект»);
- ☐ настроить физические параметры пространства, в котором будет действовать объект, — задать освещение, гравитацию, свойства атмосферы, свойства взаимодействующих объектов и поверхностей;

- ❑ задать траектории движения объектов;
- ❑ рассчитать результирующую последовательность кадров;
- ❑ наложить поверхностные эффекты на итоговый анимационный ролик.

Для создания реалистичной модели объекта используют геометрические примитивы (прямоугольник, куб, шар, конус и пр.) и гладкие, так называемые *сплайновые поверхности*. В последнем случае применяют чаще всего метод *бикубических рациональных B-сплайнов на неравномерной сетке (NURBS)*. Вид поверхности при этом определяется расположенной в пространстве сеткой опорных точек. Каждой точке присваивается коэффициент, величина которого определяет степень ее влияния на часть поверхности, проходящей вблизи точки. От взаимного расположения точек и величины коэффициентов зависит форма и «гладкость» поверхности в целом. Специальный инструментарий позволяет обрабатывать примитивы, составляющие объект, как единое целое, с учетом их взаимодействия на основе заданной физической модели.



Деформация объекта обеспечивается перемещением контрольных точек, расположенных вблизи. Каждая контрольная точка связана с близлежащими опорными точками, степень ее влияния на них определяется удаленностью. Другой метод называют *сеткой деформации*. Вокруг объекта или его части размещается трехмерная сетка, перемещение любой точки которой вызывает упругую деформацию как самой сетки, так и окруженного объекта.

Еще одним способом построения объектов из примитивов служит *твердотельное моделирование*. Объекты представлены твердыми телами, которые при взаимодействии с другими телами различными способами (объединение, вычитание, слияние и др.) претерпевают необходимую трансформацию. Например, вычитание из прямоугольного параллелепипеда шара приведет к образованию в параллелепипеде полукруглой лунки.

После формирования «скелета» объекта необходимо покрыть его поверхность материалами. Все многообразие свойств материалов в компьютерном модели-

ровании сводится к визуализации поверхности, то есть к расчету коэффициента прозрачности поверхности и угла преломления лучей света на границе материала и окружающего пространства. Для построения поверхностей материалов используют пять основных физических моделей:

- ❑ *Bouknight* — поверхности с диффузным отражением без бликов (например, матовый пластик);
- ❑ *Phong* — поверхности со структурированными микронеровностями (например, металлические);
- ❑ *Blinn* — поверхности со специальным распределением микронеровностей с учетом взаимных перекрытий (например, глянец);
- ❑ *Whitted* — модель, позволяющая дополнительно учитывать поляризацию света;
- ❑ *Hall* — модель, позволяющая корректировать направления отражения и параметры преломления света.

Закраска поверхностей осуществляется методами Гуро (*Gouraud*) или Фонга (*Phong*). В первом случае цвет примитива рассчитывается лишь в его вершинах, а затем линейно интерполируется по поверхности. Во втором случае строится нормаль к объекту в целом, ее вектор интерполируется по поверхности составляющих примитивов, и освещение рассчитывается для каждой точки.



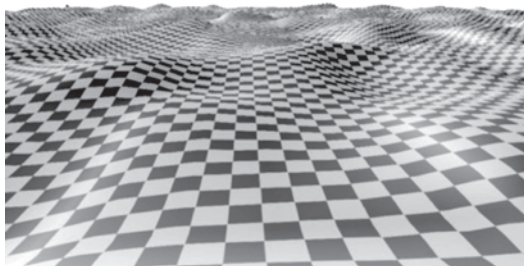
Свет, уходящий с поверхности в конкретной точке в сторону наблюдателя, представляет собой сумму компонентов, умноженных на коэффициент, связанный с материалом и цветом поверхности в данной точке. К таковым компонентам относятся:

- ❑ свет, пришедший с обратной стороны поверхности, то есть преломленный свет (*Refracted*);
- ❑ свет, равномерно рассеиваемый поверхностью (*Diffuse*);
- ❑ зеркально отраженный свет (*Reflected*);
- ❑ блики, то есть отраженный свет источников (*Specular*);
- ❑ собственное свечение поверхности (*Self Illumination*).

Свойства поверхности описываются в создаваемых массивах текстур (двух- или трехмерных). Таким образом, в массиве содержатся данные о степени прозрачности материала; коэффициенте преломления; коэффициентах смещения компонентов (их список указан выше); цвете в каждой точке, цвете блика, его ширине и резкости;

цвете рассеянного (фонового) освещения; локальных отклонениях векторов от нормали (то есть учитывается шероховатость поверхности).

Следующим этапом является наложение («проектирование») текстур на определенные участки каркаса объекта. При этом необходимо учитывать их взаимное влияние на границах примитивов. Проектирование материалов на объект — задача трудно формализуемая, она сродни художественному процессу и требует от исполнителя хотя бы минимальных творческих способностей.



Из всех параметров пространства, в котором действует создаваемый объект, с точки зрения визуализации самым важным является определение источников света. В трехмерной графике принято использовать виртуальные эквиваленты физических источников.

- ❑ Аналогом равномерного светового фона служит так называемый *растворенный свет* (*Ambient Light*). Он не имеет геометрических параметров и характеризуется только цветом и интенсивностью. Пример в природе — естественная освещенность вне видимости Солнца и Луны.
- ❑ Удаленный не точечный источник называют *удаленным светом* (*Distant Light*). Ему присваиваются конкретные геометрические параметры (координаты). Аналог в природе — Солнце.
- ❑ Точечный источник света (*Point Light Source*) равномерно испускает свет во всех направлениях и также имеет координаты. Аналог в технике — электрическая лампочка.
- ❑ Направленный источник света (*Direct Light Source*) кроме местоположения характеризуется направлением светового потока, углами полного конуса света и его наиболее яркого пятна. Аналог в технике — прожектор.

После завершения конструирования и визуализации объекта приступают к его «оживлению», то есть заданию параметров движения. Компьютерная анимация базируется на ключевых кадрах. В первом кадре объект выставляется в исходное положение. Через определенный промежуток (например, в восьмом кадре) задается новое положение объекта и так далее до конечного положения. Промежуточные значения вычисляет программа по специальному алгоритму. При этом происходит не просто линейная аппроксимация, а плавное изменение положения опорных точек объекта в соответствии с заданными условиями (рис. 15.7).

Эти условия определяются иерархией объектов (то есть законами их взаимодействия между собой), разрешенными плоскостями движения, предельными углами поворотов, величинами ускорений и скоростей. Такой подход называют методом *инверсной кинематики движения*. Он хорошо работает при моделировании механических устройств. В случае с имитацией живых объектов используют так называемые *скелетные модели*. То есть создается некий каркас, подвижный в точках, характерных для моделируемого объекта. Движения точек просчитываются предыдущим методом. Затем на каркас накладывается оболочка, состоящая из смоделированных поверхностей, для которых каркас является набором контрольных точек, то есть создается *каркасная модель*. Каркасная модель визуализируется наложением поверхностных текстур с учетом условий освещения. В ходе перемещения объекта получается весьма правдоподобная имитация движений живых существ.

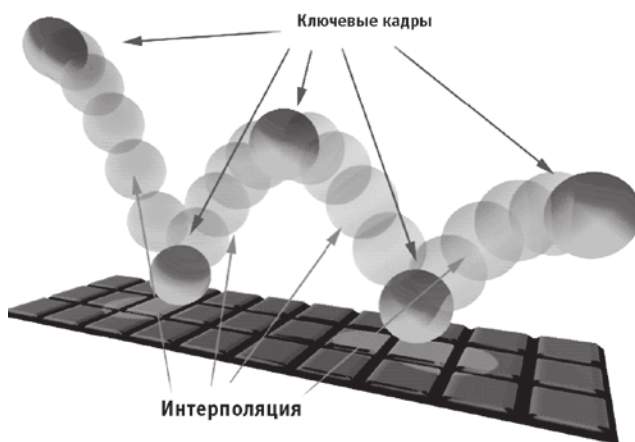


Рис. 15.7. Построение видеоряда по ключевым кадрам

Наиболее совершенный метод анимации заключается в фиксации реальных движений физического объекта. Например, на человеке закрепляют в контрольных точках яркие источники света и снимают заданное движение на видео- или кинопленку. Затем координаты точек по кадрам переводят с пленки в компьютер и присваивают соответствующим опорным точкам каркасной модели. В результате движения имитируемого объекта практически неотличимы от живого прототипа.

Процесс расчета реалистичных изображений называют *рендерингом* (визуализацией). Большинство современных программ рендеринга основаны на *методе обратной трассировки лучей* (*Backway Ray Tracing*). Его суть заключается в следующем.

1. Из точки наблюдения сцены посылается в пространство виртуальный луч, по траектории которого должно прийти изображение в точку наблюдения.
2. Для определения параметров приходящего луча все объекты сцены проверяются на пересечение с траекторией наблюдения. Если пересечения не происходит, считается, что луч попал в фон сцены и приходящая информация определяется

только параметрами фона. Если траектория пересекается с объектом, то в точке соприкосновения рассчитывается свет, уходящий в точку наблюдения в соответствии с параметрами материала.

3. Сначала просчитывается преломленный и отраженный свет, затем проверяется видимость из точки пересечения всех источников света и интенсивность светового потока. Также вычисляются наличие, резкость и ширина бликов от каждого источника света.
4. Полученные в результате итоговые значения цвета и интенсивности обрабатываются с учетом траектории луча и параметров атмосферы и присваиваются точке объекта как значения визуализации для наблюдателя. Затем процесс повторяется для всех элементов сцены. С целью упрощения расчетов пересечение проверяют не для каждой точки, а для примитива в целом. Иногда вокруг объекта создают простую виртуальную геометрическую фигуру (параллелепипед, шар), расчет пересечений для объекта выполняют только при пересечении траектории наблюдения с фигурой в целом.

Применение сложных математических моделей позволяет имитировать такие физические эффекты, как взрывы, дождь, огонь, дым, туман. Существуют методы расчета процедурных эффектов (*Procedural Effects*) и взаимодействия систем частиц (*Particle System*). Однако их применение в полном объеме требует громадных вычислительных ресурсов, и потому в персональных компьютерах обычно используют упрощенные варианты. По завершении рендеринга компьютерную трехмерную анимацию используют либо как самостоятельный продукт, либо в качестве отдельных частей или кадров готового продукта (рис. 15.8).



Рис. 15.8. Моделирование взрыва с помощью систем частиц

Особую область трехмерного моделирования в режиме реального времени составляют тренажеры технических средств — автомобилей, судов, летательных и космических аппаратов. В них необходимо очень точно реализовывать технические параметры объектов и свойства окружающей физической среды. В более простых вариантах, например при обучении вождению наземных транспортных средств, тренажеры реализуют на персональных компьютерах.

Самые совершенные на сегодняшний день устройства созданы для обучения пилотированию космических кораблей и военных летательных аппаратов. Моделированием и визуализацией объектов в таких тренажерах заняты несколько специализированных графических станций, построенных на мощных RISC-процессорах

и скоростных видеоадаптерах с аппаратными ускорителями трехмерной графики. Общее управление системой и просчет сценариев взаимодействия возложены на суперкомпьютер, состоящий из десятков и сотен процессоров. Стоимость таких комплексов выражается девятизначными цифрами, но их применение окупается достаточно быстро, так как обучение на реальных аппаратах в десятки раз дороже.

Программные средства обработки трехмерной графики

На персональных компьютерах основную долю рынка программных средств обработки трехмерной графики занимают три пакета. Эффективней всего они работают на самых мощных машинах под управлением операционных систем Microsoft Windows или Apple Macintosh.

В частности, речь идет о программных пакетах Autodesk 3ds Max (ранее продукт назывался 3D Studio MAX), Autodesk Softimage (ранее – Softimage 3D) и Autodesk Maya (ранее он назывался Alias Maya или просто Maya). Все эти пакеты, как можно заметить, принадлежат компании Autodesk. Ранее эти программы были созданы и развивались благодаря другим компаниям, однако Autodesk купила эти компании и теперь все три популярных программы для 3D-моделирования принадлежат ей.

Каждый из перечисленных программных продуктов обладает возможностями для применения в практически любой сфере современного трехмерного моделирования. Однако исторически сложились следующие области применения этих программ:

- ❑ Autodesk 3ds Max — компьютерные игры, архитектура и строительство, производство видеопродукции.



Рис. 15.9. Трехмерное моделирование средствами Autodesk 3ds Max

- ❑ Autodesk Maya — производство кинофильмов, мультфильмов.
- ❑ Autodesk Softimage — компьютерные игры, видеопродукция.

Помимо вышеперечисленных продуктов для обработки трехмерной графики, существует множество альтернативных коммерческих и бесплатных программ. Например, популярностью пользуется **Blender** – бесплатная программа, распространяемая по лицензии GNU GPL.

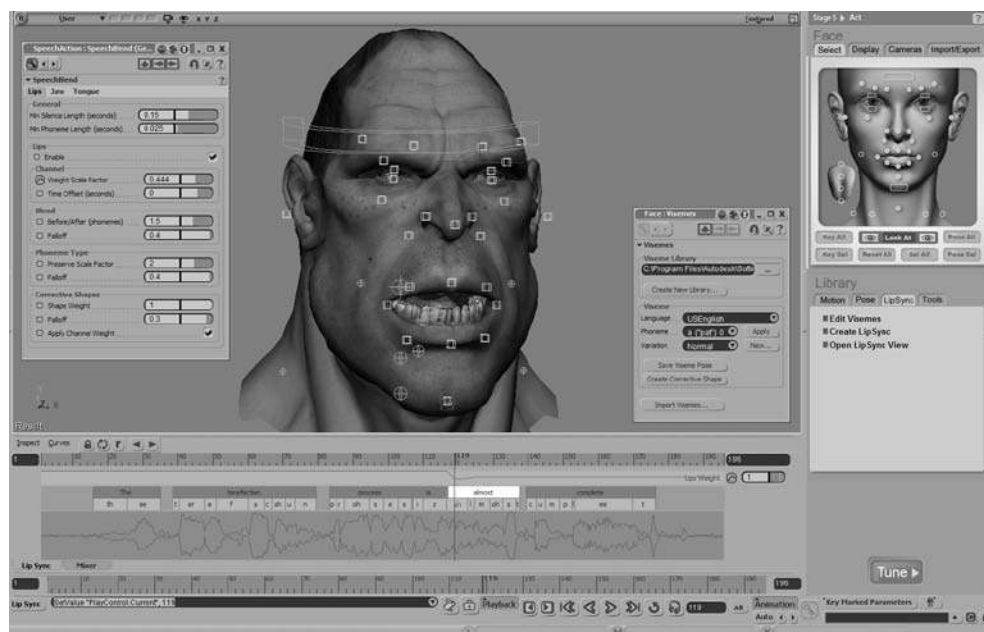


Рис. 15.10. Моделирование средствами пакета Maya

15.2. Представление графических данных

Форматы графических данных

В компьютерной графике применяют по меньшей мере три десятка форматов файлов для хранения изображений. Но лишь часть из них стала стандартом «де-факто» и применяется в подавляющем большинстве программ. Как правило, несовместимые форматы имеют файлы растровых, векторных, трехмерных изображений, хотя существуют форматы, позволяющие хранить данные разных классов. Многие приложения ориентированы на собственные «специфические» форматы, перенос их файлов в другие программы вынуждает использовать специальные фильтры или экспортировать изображения в «стандартный» формат.

TIFF (Tagged Image File Format). Формат предназначен для хранения растровых изображений высокого качества (расширение имени файла .TIF). Относится к числу широко распространенных, отличается переносимостью между платформами (IBM PC и Apple Macintosh), обеспечен поддержкой со стороны большинства

графических, верстальных и дизайнерских программ. Предусматривает широкий диапазон цветового охвата — от монохромного черно-белого до 32-разрядной модели цветоделения CMYK. Начиная с версии 6.0, в формате TIFF можно хранить сведения о масках (контурах обтравки) изображений. Для уменьшения размера файла применяется встроенный алгоритм сжатия LZW.

PSD (PhotoShop Document). Собственный формат программы Adobe Photoshop (расширение имени файла .PSD), один из наиболее мощных по возможностям хранения растровой графической информации. Позволяет запоминать параметры слоев, каналов, степени прозрачности, множества масок. Поддерживаются 48-разрядное кодирование цвета, цветоделение и различные цветовые модели. Основным недостатком выражен в том, что отсутствие эффективного алгоритма сжатия информации приводит к большому объему файлов.

Windows Bitmap. Формат хранения растровых изображений в операционной системе Windows (расширение имени файла .BMP). Соответственно, поддерживается всеми приложениями, работающими в этой среде.

JPEG (Joint Photographic Experts Group). Формат предназначен для хранения растровых изображений (расширение имени файла .JPG). Позволяет регулировать соотношение между степенью сжатия файла и качеством изображения. Применяемые методы сжатия основаны на удалении «избыточной» информации, поэтому формат рекомендуют использовать только для электронных публикаций.

GIF (Graphics Interchange Format). Стандартизирован в 1987 году как средство хранения сжатых изображений с фиксированным (256) количеством цветов (расширение имени файла .GIF). Получил популярность в Интернете благодаря высокой степени сжатия. Последняя версия формата GIF89a позволяет выполнять чересстрочную загрузку изображений и создавать рисунки с прозрачным фоном. Ограниченные возможности по количеству цветов обуславливают его применение исключительно в электронных публикациях. Кроме того, формат GIF поддерживает создание анимированных изображений. До появления более совершенных средств веб-анимации этот формат был основным средством для создания анимированных веб-страниц.

PNG (Portable Network Graphics). Сравнительно новый (1995) формат хранения изображений, предназначенный для их публикации в Интернете (расширение имени файла .PNG). Создавался как замена для форматов GIF и JPEG. Поддерживаются три типа изображений — цветные с глубиной 8 или 24 бита и черно-белое с градацией 256 оттенков серого. Сжатие информации происходит практически без потерь, предусмотрены 254 уровня альфа-канала, чересстрочная развертка. В настоящий момент используется, преимущественно для целей веб-публикаций наряду с теряющим популярность GIF и широко распространенным JPG.

WMF (Windows MetaFile). Формат хранения векторных изображений операционной системы Windows (расширение имени файла .WMF). По определению поддерживается всеми приложениями этой системы. Однако отсутствие средств для работы со стандартизированными цветовыми палитрами, принятыми в полиграфии, и другие недостатки ограничивают его применение.

EPS (Encapsulated PostScript). Формат описания как векторных, так и растровых изображений на языке PostScript фирмы Adobe, фактическом стандарте

в области допечатных процессов и полиграфии (расширение имени файла .EPS). Так как язык PostScript является универсальным, в файле могут одновременно храниться векторная и растровая графика, шрифты, контуры обтравки (маски), параметры калибровки оборудования, цветовые профили. Для отображения на экране векторного содержимого используется формат WMF, а растрового — TIFF. Но экранная копия лишь в общих чертах отображает реальное изображение, что является существенным недостатком EPS. Действительное изображение можно увидеть лишь на выходе выводного устройства, с помощью специальных программ просмотра или после преобразования файла в формат PDF в приложении Adobe Acrobat Reader или в подобных ему.

PDF (Portable Document Format). Формат описания документов, разработанный фирмой Adobe (расширение имени файла .PDF). Хотя этот формат в основном предназначен для хранения документа целиком, его впечатляющие возможности позволяют обеспечить эффективное представление изображений. Формат является аппаратно-независимым, поэтому вывод изображений допустим на любых устройствах — от экрана монитора до фотоэкспонирующего устройства. Мощный алгоритм сжатия со средствами управления итоговым разрешением изображения обеспечивает компактность файлов при высоком качестве иллюстраций.

Понятие цвета

Цвет чрезвычайно важен в компьютерной графике как средство усиления зрительного впечатления и повышения информационной насыщенности изображения. Ощущение цвета формируется человеческим мозгом в результате анализа светового потока, попадающего на сетчатку глаза от излучающих или отражающих объектов. Считается, что цветовые рецепторы (колбочки) подразделяются на три группы, каждая из которых воспринимает только единственный цвет — красный, зеленый или синий. Нарушения в работе любой из групп приводит к явлению *дальтонизма* — искаженного восприятия цвета.

Световой поток формируется излучениями, представляющими собой комбинацию трех «чистых» спектральных цветов (красный, зеленый, синий — КЗС) и их производных (в англоязычной литературе используют аббревиатуру *RGB* — *Red, Green, Blue*). Для излучающих объектов характерно *аддитивное цветовоспроизведение* (световые излучения суммируются), для отражающих объектов — *субтрактивное цветовоспроизведение* (световые излучения вычитаются). Примером объекта первого типа является электронно-лучевая трубка монитора, второго типа — полиграфический отпечаток.

Физические характеристики светового потока определяются параметрами *мощности, яркости и освещенности*. Визуальные параметры ощущения цвета характеризуются *светлотой*, то есть различимостью участков, сильнее или слабее отражающих свет. Минимальную разницу между яркостью различимых по светлоте объектов называют *порогом*. Величина порога пропорциональна логарифму отношения яркостей. Последовательность оптических характеристик объекта (расположенная по возрастанию или убыванию), выраженная в оптических плотностях или логарифмах яркостей, составляет *градицию* и является важнейшим инструментом для анализа и обработки изображения.

Для точного цветовоспроизведения изображения на экране монитора важным является понятие *цветовой температуры*. В классической физике считается, что любое тело с температурой, отличной от 0 градусов по шкале Кельвина, испускает излучение. С повышением температуры спектр излучения смещается от инфракрасного до ультрафиолетового диапазона, проходя через оптический.

Для идеального черного тела легко находится зависимость между длиной волны излучения и температурой тела. На основе этого закона, например, была дистанционно вычислена температура Солнца — около 6500 К. Для целей правильного цветовоспроизведения характерна обратная задача. То есть монитор с выставленной цветовой температурой 6500 К должен максимально точно воспроизвести спектр излучения идеального черного тела, нагретого до такой же степени. Таким образом, стандартные значения цветовых температур используют в качестве всеобщего эталона, обеспечивающего одинаковое цветовоспроизведение на разных излучающих устройствах.

На практике зрение человека непрерывно подстраивается под спектр, характерный для цветовой температуры источника излучения. Например, на улице в яркий солнечный день цветовая температура составляет около 7000 К. Если с улицы зайти в помещение, освещенное только лампами накаливания (цветовая температура около 2800 К), то в первый момент свет ламп покажется желтым, белый лист бумаги тоже приобретет желтый оттенок. Затем происходит адаптация зрения к новому соотношению КЗС, характерному для цветовой температуры 2800 К, и свет лампы и лист бумаги начинают восприниматься как белые.

Насыщенность цвета показывает, насколько данный цвет отличается от монохроматического («чистого») излучения того же цветового тона. В компьютерной графике за единицу принимается насыщенность цветов спектральных излучений.

Ахроматические цвета (белый, серый, черный) характеризуются только светлотой. *Хроматические цвета* имеют параметры насыщенности, светлоты и цветового тона.

Способы описания цвета

В компьютерной графике применяют понятие *цветового разрешения* (другое название — *глубина цвета*). Оно определяет метод кодирования цветовой информации для ее воспроизведения на экране монитора. Для отображения черно-белого изображения достаточно двух бит (белый и черный цвета). Восемьразрядное кодирование позволяет отобразить 256 градаций цветового тона. Два байта (16 бит) определяют 65 536 оттенков (такой режим называют *High Color*). При 24-разрядном способе кодирования возможно определить более 16,5 миллиона цветов (режим называют *True Color*).

С практической точки зрения цветовому разрешению монитора близко понятие *цветового охвата*. Под ним подразумевается диапазон цветов, который можно воспроизвести с помощью того или иного устройства вывода (монитор, принтер, печатная машина и пр.).

В соответствии с принципами формирования изображения аддитивным или субтрактивным методами разработаны способы разделения цветового оттенка на составляющие компоненты, называемые *цветовыми моделями*. В компьютерной

графике в основном применяют модели *RGB* и *HSB* (для создания и обработки аддитивных изображений) и *СМУК* (для печати копии изображения на полиграфическом оборудовании).

Цветовые модели расположены в трехмерной системе координат, образующей *цветовое пространство*, так как из *законов Грассмана* следует, что цвет можно выразить точкой в трехмерном пространстве.

Первый закон Грассмана (закон трехмерности). *Любой цвет однозначно выражается тремя составляющими, если они линейно независимы. Линейная независимость заключается в невозможности получить любой из этих трех цветов сложением двух остальных.*

Второй закон Грассмана (закон непрерывности). *При непрерывном изменении излучения цвет смеси также меняется непрерывно. Не существует такого цвета, к которому нельзя было бы подобрать бесконечно близкий.*

Третий закон Грассмана (закон аддитивности). *Цвет смеси излучений зависит только от их цвета, но не спектрального состава.* То есть цвет (*C*) смеси выражается суммой цветовых уравнений излучений:

$$C_1 = R_1R + G_1G + B_1B;$$

$$C_2 = R_2R + G_2G + B_2B;$$

$$C_n = R_nR + G_nG + B_nB;$$

$$C_{\text{сумм}} = (R_1 + R_2 + \dots + R_n) R + (G_1 + G_2 + \dots + G_n) G + (B_1 + B_2 + \dots + B_n) B.$$

Таким образом, прямоугольная трехмерная координатная система цветового пространства для аддитивного способа формирования изображения имеет точку начала координат, соответствующую абсолютно черному цвету (цветовое излучение отсутствует), и три оси координат, соответствующих основным цветам. Любой цвет (*C*) может быть выражен в цветовом пространстве вектором, который описывается уравнением:

$$\vec{C}_n = R_n \vec{R} + G_n \vec{G} + B_n \vec{B},$$

которое практически идентично уравнению свободного вектора в пространстве, рассматриваемому в векторной алгебре. Направление вектора характеризует цветность, а его модуль выражает яркость.

Так как величина излучения основных цветов является основой цветовой модели, ее максимальное значение принято считать за единицу. Тогда в трехмерном цветовом пространстве можно построить *плоскость единичных цветов*, образованную *треугольником цветности*. Каждой точке плоскости единичных цветов соответствует след цветового вектора, пронизывающего ее в этой точке (рис. 15.11). Следовательно, цветность любого излучения может быть представлена единственной точкой внутри треугольника цветности, в вершинах которого находятся точки основных цветов. То есть положение точки любого цвета можно задать двумя координатами, а третья легко находится по двум другим.

Если на плоскости единичных цветов указать значения координат, соответствующих реальным спектральным излучениям оптического диапазона (от 380 до

700 нм), и соединить их кривой, то мы получим линию, являющуюся геометрическим местом точек цветности монохроматических излучений, называемую *локусом*. Внутри локуса находятся все реальные цвета (рис. 15.12).

Чтобы избежать отрицательных значений координат, была выбрана колориметрическая система XYZ , полученная путем пересчета из RGB . В этой системе точке белого соответствуют координаты $(0,33; 0,33)$. Колориметрическая система XYZ является универсальной, в ней можно выразить цветовой охват как аддитивных, так и субтрактивных источников цвета. Для аддитивных источников цветовой охват выражается треугольником с координатами вершин, соответствующими излучению основных цветов R, G, B .

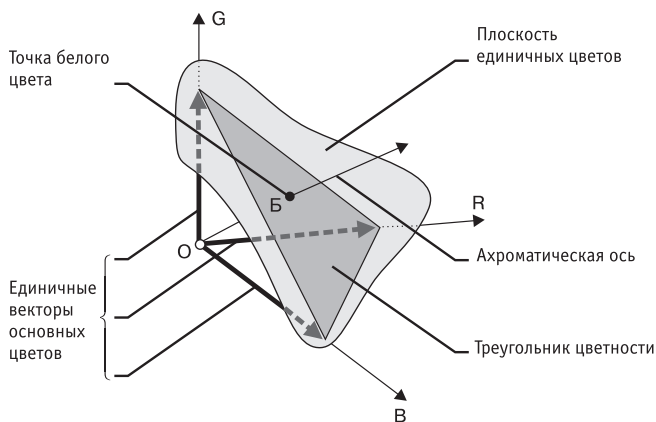


Рис. 15.11. Плоскость единичных цветов

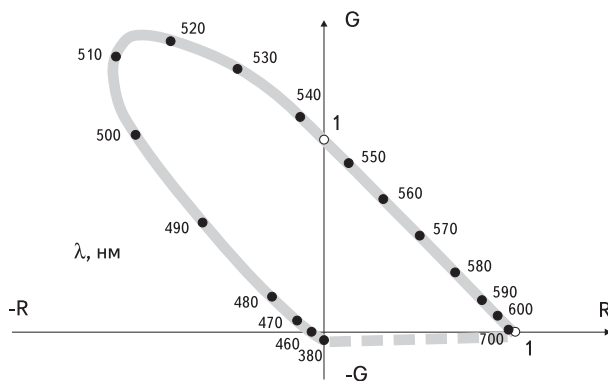


Рис. 15.12. Цветовой локус

Для субтрактивных источников (полученных в процессе печати красками, чернилами, красителями) используется модель СМΥК, поэтому цветовой охват описывается шестиугольником, когда помимо точек синтеза основной триады (желтая, пурпурная, голубая) добавляются точки попарных наложений, соответствующие

основным цветам: желтая + голубая $\times$ зеленая, желтая + пурпурная $\times$ красная, голубая + пурпурная $\times$ синяя (рис. 15.13).

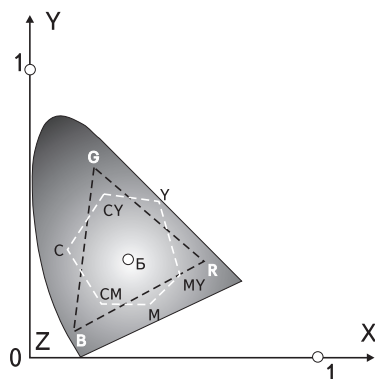


Рис. 15.13. Колориметрическая система XYZ

Цветовая модель CIE Lab

В 1920 году была разработана цветовая пространственная модель *CIE Lab* (*Communication Internationale de l'Eclairage* — международная комиссия по освещению; L, a, b — обозначения осей координат в этой системе). Система является аппаратно независимой и потому часто применяется для переноса данных между устройствами. В модели CIE Lab любой цвет определяется светлотой (L) и хроматическими компонентами: параметром a , изменяющимся в диапазоне от зеленого до красного, и параметром b , изменяющимся в диапазоне от синего до желтого. Цветовой охват модели CIE Lab значительно превосходит возможности мониторов и печатных устройств, поэтому перед выводом изображения, представленного в этой модели, его приходится преобразовывать. Данная модель была разработана для согласования цветных фотохимических процессов с полиграфическими. Сегодня она является принятым по умолчанию стандартом для программы Adobe Photoshop.

Цветовая модель RGB

Цветовая модель *RGB* является аддитивной, то есть любой цвет представляет собой сочетание в различной пропорции трех основных цветов — красного (*Red*), зеленого (*Green*), синего (*Blue*). Она служит основой при создании и обработке компьютерной графики, предназначенной для электронного воспроизведения (на мониторе, телевизоре). При наложении одного компонента основного цвета на другой яркость суммарного излучения увеличивается. Совмещение трех компонентов одинаковой яркости дает ахроматический серый цвет, который при увеличении яркости приближается к белому цвету. При 256 градационных уровнях тона черному цвету соответствуют нулевые значения RGB, а белому — максимальные, с координатами (255, 255, 255).

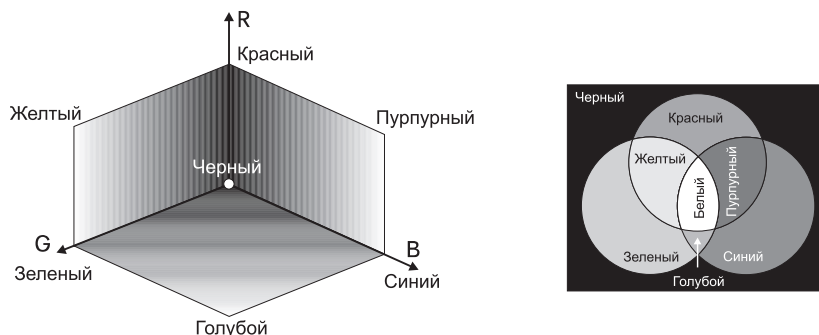


Рис. 15.14. Аддитивная цветовая модель RGB

Цветовая модель HSB

Цветовая модель *HSB* разработана с максимальным учетом особенностей восприятия цвета человеком. Она построена на основе цветового круга Манселла. Цвет описывается тремя компонентами: оттенком (*Hue*), насыщенностью (*Saturation*) и яркостью (*Brightness*). Значение цвета выбирается как вектор, исходящий из центра окружности. Точка в центре соответствует белому цвету, а точки по периметру окружности — чистым спектральным цветам. Направление вектора задается в градусах и определяет цветовой оттенок. Длина вектора определяет насыщенность цвета. На отдельной оси, называемой *ахроматической*, задается яркость, при этом нулевая точка соответствует черному цвету. Цветовой охват модели HSB перекрывает все известные значения реальных цветов.

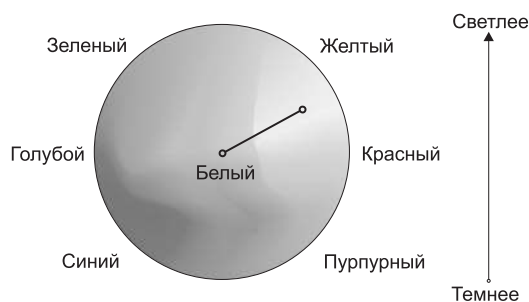


Рис. 15.15. Цветовая модель HSB

Модель HSB принято использовать при создании изображений на компьютере с имитацией приемов работы и инструментария художников. Существуют специальные программы, имитирующие кисти, перья, карандаши. Обеспечивается имитация работы с красками и различными полотнами. После создания изображения его рекомендуется преобразовать в другую цветовую модель, в зависимости от предполагаемого способа публикации.

Цветовая модель СМУК, цветоделение

Цветовая модель *СМУК* относится к субтрактивным и ее используют при подготовке публикаций к печати. Цветовыми компонентами *СМУ* служат цвета, полученные вычитанием основных из белого:

- ❑ голубой (*cyan*) = белый – красный = зеленый + синий;
- ❑ пурпурный (*magenta*) = белый – зеленый = красный + синий;
- ❑ желтый (*yellow*) = белый – синий = красный + зеленый.

Такой метод соответствует физической сущности восприятия отраженных от печатных оригиналов лучей. Голубой, пурпурный и желтый цвета называются *дополнительными*, потому что они дополняют основные цвета до белого. Отсюда вытекает и главная проблема цветовой модели *СМУ* — наложение друг на друга дополнительных цветов на практике не дает чистого черного цвета. Поэтому в цветовую модель был включен компонент чистого черного цвета. Так появилась четвертая буква в аббревиатуре цветовой модели *СМУК* (*Cyan, Magenta, Yellow, black*).

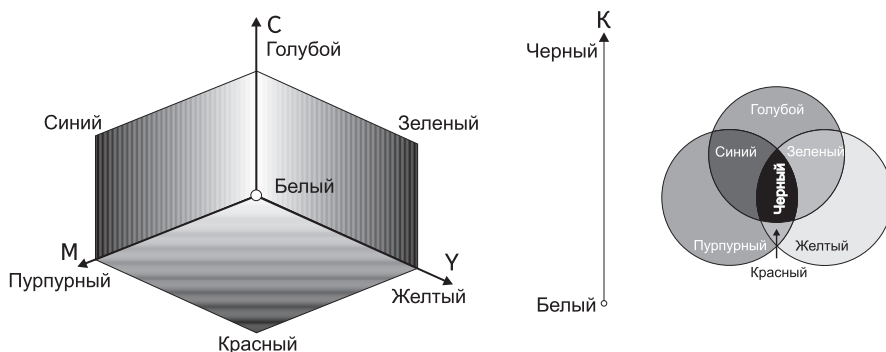


Рис. 15.16. Цветовая модель СМУК

Для печати на полиграфическом оборудовании цветное компьютерное изображение необходимо разделить на составляющие, соответствующие компонентам цветовой модели *СМУК*. Этот процесс называют *цветоделением*. В итоге получают четыре отдельных изображения, содержащих одноцветное содержимое каждого компонента в оригинале. Затем в типографии с форм, созданных на основе цветоделенных пленок, печатают многоцветное изображение, получаемое наложением цветов *СМУК*.

Цветовая палитра

Электронная *цветовая палитра* в компьютерной графике по предназначению подобна палитре художника, но включает гораздо большее число цветов. Электронная палитра состоит из определенного числа ячеек, каждая из которых содержит отдельный цветовой тон. Конкретная цветовая палитра соотносится с определенной цветовой моделью, так как ее цвета созданы на основе цветового пространства этой модели. Но если в цветовой модели возможно воспроизвести любой из опи-

сываемых ею цветов, цветовая палитра содержит ограниченный набор цветов, называемых *стандартными*.

Примером стандартных цветовых палитр являются наборы фирмы *Pantone*, ориентированные на полиграфическую публикацию изображений. Программы создания и обработки компьютерной графики, как правило, предоставляют на выбор несколько цветовых палитр в цветовых моделях *RGB*, *HSB*, *CIE Lab*, *CMYK*.

Состав цветовых палитр *RGB* зависит от выбранного цветового разрешения — 24, 16 или 8 бит. В последнем случае цветовая палитра называется *индексной*, потому что каждый цветовой оттенок кодируется одним числом, которое выражает не цвет пиксела, а *индекс* (номер) цвета. Таким образом, к файлу цветного изображения, созданного в индексной палитре, должна быть приложена сама палитра, так как программе обработки компьютерной графики неизвестно, какая именно палитра была использована.

Изображения, подготавливаемые для публикации в Интернете, принято создавать в так называемой *безопасной палитре* цветов. Она является вариантом рассмотренной выше индексной палитры. Но так как файлы изображений в веб-графике должны иметь минимальный размер, необходимо было отказаться от включения в их состав индексной палитры. Для этого была принята единая фиксированная палитра цветов, названная «*безопасной*», то есть обеспечивающей правильное отображение цветов на любых устройствах (в программах), поддерживающих единую палитру. Безопасная палитра содержит всего 216 цветов, что связано с ограничениями, накладываемыми требованиями совместимости с компьютерами, не относящимися к классу IBM PC.

Системы управления цветом

При создании и обработке элементов компьютерной графики необходимо добиться, чтобы изображение выглядело практически одинаково на всех стадиях процесса, на любом устройстве отображения, при любом методе визуализации (аддитивном или субтрактивном). Иначе, чем больше переходных этапов будет содержать процесс обработки, тем большие искажения будут вноситься в оригинал, и конечный результат может совершенно не удовлетворять даже минимальным требованиям к качеству. Для согласования цветов на всех стадиях обработки компьютерной графики применяют *системы управления цветом* (*Color Management System — CMS*).

Такие системы содержат набор объективных параметров, обязательных для всех устройств при обмене цветовыми данными. Универсальность *CMS* достигается введением трех типов переменных, каждая из которых управляет представлением цвета на своем уровне:

Цветовая гамма. Каждый тип устройства имеет свою *цветовую гамму*, область которой всегда меньше, чем цветовой охват практически любой цветовой модели. *CMS* управляет преобразованием цвета между различными цветовыми моделями с учетом цветовой гаммы конкретных устройств.

Профиль. Каждое устройство воспроизводит цвета особым образом, что зависит от технических и программных решений, принятых изготовителем. Для согласования отображения цветов на различных устройствах они должны иметь

собственный *профиль*, описывающий различия в представлении цвета между устройством и определенной цветовой моделью. Международным консорциумом по цвету (*International Color Consortium — ICC*) установлен промышленный стандарт на параметры описания характеристик воспроизведения цвета. Устройства, имеющие профиль ICC, напрямую управляются CMS. В противном случае возможна генерация профиля в некоторых системах CMS.

Калибровка. Даже устройства одной модели от одного производителя имеют отличия в реализации профиля ICC, обусловленные допусками при изготовлении компонентов, условиями эксплуатации, внешними помехами. Поэтому CMS, как правило, включает средства *калибровки*, то есть настройки конкретного экземпляра в соответствии с требованиями профиля ICC и фиксации неустранимых отклонений (с целью их программной компенсации). Средства калибровки могут быть аппаратно-программными и чисто программными. Сам процесс калибровки выполняется с периодичностью, установленной изготовителем, или автоматически, при выходе параметров ICC за границы допусков.

Не существует идеальной системы управления цветом, одинаково пригодной для всех устройств, одинаково работающей на всех платформах и во всех программных средах. Наиболее близко к идеалу подходят CMS, реализованные на уровне операционной системы. Впервые CMS под названием ColorSync в операционную систему встроила фирма Apple, что предопределило успех компьютеров Macintosh в сфере издательской деятельности, допечатной подготовки и полиграфии. Операционная система Windows 7 поддерживает два основных цветовых профиля: цветовую систему Windows (WCS) и профиль международного консорциума по цвету (ICC).

Из CMS, являющихся внешними по отношению к операционной системе, наибольшее распространение получили программы фирм, давно работающих в области цветной фотографии, печати, цифровых графических технологий.

Agfa Foto Tune. Эта система управления цветом работает на платформах Windows и Apple. Включает множество профилей ICC для мониторов, цветных принтеров, сканеров, цифровых фотокамер, полиграфического оборудования. Имеются средства создания заказных профилей для устройств, не попавших в список. Преобразования между цветовыми профилями устройств (например, сканер — монитор) могут производиться напрямую, без промежуточного конвертирования в цветовую модель CIE Lab и обратно.

Kodak DayStar ColorMatch. Система предназначена для пользователей пакетов Adobe Photoshop и QuarkXPress. Отличается модульным построением, поэтому базовая поставка содержит ограниченное число профилей, а остальные необходимо приобретать дополнительно. Система имеет средства поддержки формата Kodak PhotoCD с учетом вывода изображений на фотопринтеры. Средства калибровки включают стандартный шаблон IT8 для сканеров и устройство Digital Colorimeter для мониторов.

Практическое занятие

Упражнение 15.1. Расчет требуемых линиатур растра

Рассчитать, какая линиатура потребуется для печати изображения на черно-белом принтере с разрешением 1200 dpi при требовании к качеству 100 уровней серого цвета. Можно ли распечатать на таком принтере изображение с 256 градациями тона?

Подставим исходные параметры в формулу $l_{pi} = \frac{dpi}{\sqrt{N \times 1}}$:

$$l_{pi} = \frac{1200}{\sqrt{100 \times 1}} = \frac{1200}{\sqrt{99}} = \frac{1200}{9,94} = 120,72.$$

1. Так как линиатуру лучше представлять целым числом, округляем полученное значение в меньшую сторону и получаем значение 120 lpi.
2. Проверим, можно ли вывести изображение с 256 градациями тона:

$$l_{pi} = \frac{1200}{\sqrt{256 \times 1}} = \frac{1200}{\sqrt{255}} = \frac{1200}{15,96} = 75,18.$$

3. Округляем результат до ближайшего целого, получаем 75 lpi. Полученный оттиск может быть использован для размножения полиграфическими средствами с качеством, соответствующим газетному производству.

Итог. При требовании к уровню качества 100 градаций тона возможна печать с линиатурой 120 lpi, характерной для книжно-журнального производства. При воспроизведении оригинала с 256 градациями серого возможна печать с линиатурой 75 lpi, характерной для газетного производства.

Упражнение 15.2. Расчет требуемого разрешения оцифровки

Рассчитать требуемое разрешение оцифровки цветного 35-мм слайда, если предполагается печать его цветной копии размером А3 альбомного формата с линиатурой растра 133 lpi. Справка: ширина листа А3 альбомной ориентации равна 420 мм.

1. Сначала определим коэффициент масштабирования:

$$420 : 35 = 12.$$

2. Определим исходное разрешение оцифровки:

$$dpi = 133 \cdot 1,5 = 199,5, \text{ округленно} = 200 \text{ dpi}.$$

3. Вычисляем разрешение оцифровки с учетом масштабирования:

$$200 \cdot 12 = 2400 \text{ dpi}.$$

4. Устанавливаем окончательное разрешение оцифровки с учетом угла поворота растра:

$$2400 \cdot 1,06 = 2540 \text{ dpi.}$$

Для качественного полиграфического воспроизведения изображений с цветных слайдов необходимо использовать сканеры профессионального класса, обеспечивающие разрешение не ниже 2540 dpi.

Упражнение 15.3. Расчет угла поворота растра

Вам поручено подготовить на компьютере цветоделенные изображения для вывода на фотоавтомате, с учетом печати двумя дополнительными применительно к модели СМУК красками — зеленой и синей. Указать углы поворота растра для всех пленок.

1. Выберем для модели СМУК традиционные углы поворота растра, которые и подставим в таблицу.
2. Зеленый цвет является в модели СМУК производным, то есть суммой желтого и голубого, поэтому прибавим стандартный угол поворота 45° к значению угла для желтого цвета:

$$90 + 45 = 135^\circ.$$

3. Синий цвет также является в модели СМУК производным, то есть суммой голубого и пурпурного. Прибавим стандартный угол поворота к значению угла для голубого цвета:

$$105 + 45 = 150^\circ.$$

Печатная форма для краски	Угол поворота растра, градусов
Голубая	105
Пурпурная	75
Желтая	90
Черная	45
Зеленая	135
Синяя	150

В полиграфии возможна печать числом красок более четырех (из стандартного набора СМУК). Такой прием позволяет добиться на отпечатке ярких, сочных тонов именно того цвета, который необходимо подчеркнуть.

15.3. Средства для работы с растровой графикой

Программные средства создания растровых изображений

Среди программ, предназначенных для создания компьютерной двумерной живописи и графики, самыми популярными считаются Corel Painter и Adobe Photoshop.

Corel Painter рассчитан на создание изображений «с чистого листа» и ориентирован, преимущественно, на художников. Adobe Photoshop рассчитан в основном на фотографов, которые хотят обрабатывать существующие изображения. В то же время Photoshop вполне можно использовать для создания собственных изображений.

Аппаратные средства получения растровых изображений

К аппаратным средствам получения цифровых растровых оригиналов в основном относятся *сканеры* и *цифровые фотокамеры*. Другие устройства, например цифровые видеокамеры, адаптеры захвата телевизионных кадров, в компьютерной графике играют чаще вспомогательную роль. Для создания изображений «от руки» предназначены *графические планшеты*, на которых рисуют специальным электронным пером.

Сканеры по способу восприятия изображения делятся на две группы: устройства с электронными фотоумножителями (ФЭУ) и устройства на приборах с зарядовой связью (ПЗС, английская аббревиатура CCD). Сканеры с фотоумножителями называют *барабанными* — внутри аппарата помещен прозрачный барабан, на который крепится оригинал (отражающий или просветный). Затем барабан начинает вращаться с большой скоростью. Сканирующая головка имеет мощный источник света с фокусированным лучом и ФЭУ, которые движутся вдоль продольной оси барабана. Отраженный или проходящий световой поток попадает на ФЭУ (обычно имеется по одному ФЭУ на каждый канал) через прецизионную зеркальную систему развертки. Накопленный ФЭУ заряд преобразуется в цифровое значение аналого-цифровым преобразователем высокой разрядности. Так как процесс до этого момента по сути аналоговый, удается добиться очень высоких значений динамического диапазона. То есть оригинал правильно оцифровывается и в светлых, и в темных участках. Выходное разрешение оригинала достигает 5000–6000 точек на дюйм. За совершенное качество приходится платить — барабанные сканеры чрезвычайно дорогостоящи и требовательны к условиям эксплуатации.

Прочие сканеры относятся к *устройствам на ПЗС*. В отличие от ФЭУ приборы с зарядовой связью представляют собой фотоприемник, выполненный на кремниевых элементах, объединенных в линейку. Каждый светочувствительный элемент обладает способностью накапливать заряды пропорционально числу попавших на него фотонов. За время экспозиции возникает матрица зарядов, пропорциональных яркости исходного изображения. По вертикали развертка осуществляется передвижением либо всей линейки ПЗС с помощью шагового электродвигателя, либо перемещением оригинала.

Разрешающая способность определяется числом оптических элементов на единицу длины. В устройствах бытового класса это 300–600 элементов на дюйм, профессионального — 1200–3000. Программная интерполяция оптического разрешения никакого реального повышения качества оцифровки не дает. Динамический диапазон устройств на ПЗС ниже, чем у ФЭУ, потому что кремниевые элементы имеют худшее соотношение сигнал/шум.

В высокоточных сканерах на ПЗС дополнительно применяются: система зеркальной развертки по обеим координатам с компенсацией искажений по краям

оригинала, несколько линеек ПЗС, стабильные по цветовой температуре осветительные лампы, многозарядные цифро-аналоговые преобразователи, элементы, выполненные на CMOS-пластинах. Такие устройства по качеству оцифровки приближаются к барабанным сканерам, а по стоимости значительно доступнее.

Конструктивно барабанные сканеры выполняют с вертикальным или горизонтальным барабаном, съёмным или несъёмным. Сканеры на ПЗС бывают листовые, планшетные, проекционные, ручные и так называемые слайдовые (для сканирования оригиналов «на просвет»).

Для целей компьютерной графики важно не столько разрешение сканера (оно может не превышать 300 dpi), сколько хороший динамический диапазон. Для сканирования в отраженном свете желательно иметь динамический диапазон не ниже 2, «на просвет» — не ниже 3,5.

Основой *цифровых фотокамер* служит матрица ПЗС, состоящая из двумерного массива элементов. Для целей электронной публикации и непрофессионального применения достаточное число элементов на матрице около 8 миллионов. Полупрофессиональные камеры должны иметь разрешение матрицы не ниже 10 миллионов элементов, профессиональные аппараты — не менее 12 миллионов. Оцифрованные с их помощью изображения можно использовать для подготовки полиграфических публикаций. Оптическая система цифровых камер профессионального класса должна обеспечивать разрешение не ниже 110–120 пар линий на дюйм.

Графические планшеты представляют собой координатную двумерную электронную сетку, каждый элемент которой способен воспринимать и передавать ряд сигналов от электронного пера. К таковым сигналам относятся: координаты точки контакта пера с планшетом, сила нажима, угол наклона, скорость прохода (то есть время экспозиции) и ряд других. Затем за счет программного преобразования полученные данные отображаются на экране в виде линий, мазков и других художественных средств создания изображений. Обладая достаточным навыком работы с графическим планшетом, удастся очень точно имитировать различную живописную технику — письмо маслом, рисунок углем, аэрографом, карандашом и т. д.

Программа обработки растровой графики Adobe Photoshop

В обширном классе программ для обработки растровой графики особое место занимает пакет Photoshop компании Adobe. По сути дела, сегодня он является стандартом в компьютерной графике, и все другие программы неизменно сравнивают именно с ним (рис. 15.17).

Главные элементы управления программы Adobe Photoshop сосредоточены в строке меню, панели инструментов и панели свойств. Особую группу составляют диалоговые окна — *инструментальные палитры*. Далее мы рассмотрим функции перечисленных средств.

Первичное получение оригинала происходит через меню Файл командой Открыть либо одной из других команд, в частности Открыть как, Импортировать. *Импортом* называют получение изображения из внешнего источника. В частности, импорт изображений можно произвести со сканера или камеры, из видеофайла. Импорт изображений из PDF-файлов производится средствами команды Открыть.

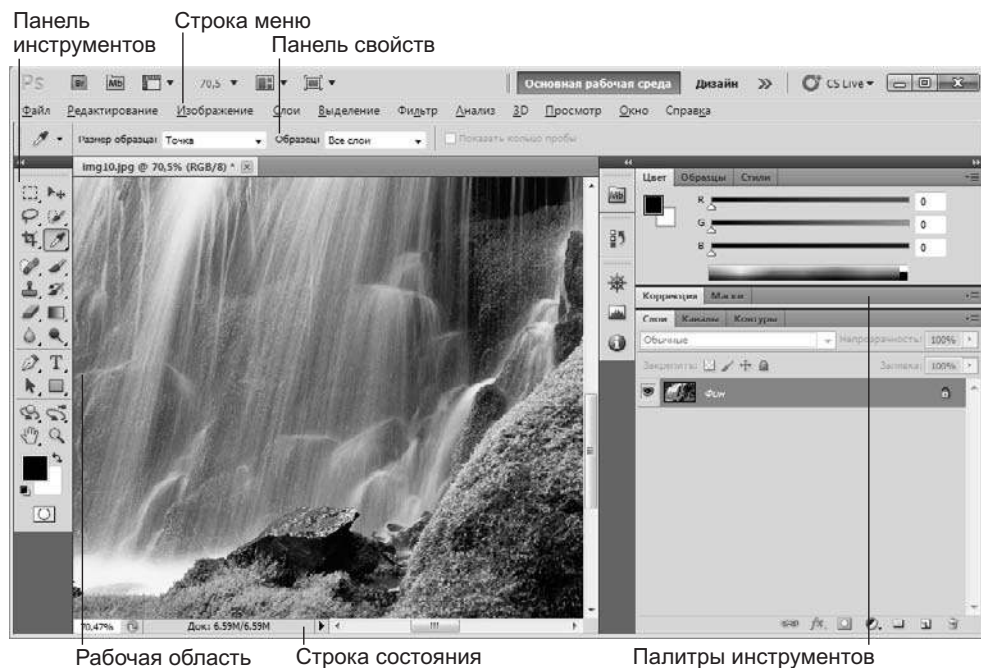


Рис. 15.17. Рабочее окно графического редактора Adobe Photoshop

Прежде чем начать операции с оригиналом изображения, следует уяснить его параметры. Для этого командой **Изображение ► Размер изображения** открывают диалоговое окно **Размер изображения**. В группах **Размерность** и **Размер печатного оттиска** приведены ширина и высота оригинала в пикселах и сантиметрах соответственно, а также разрешение (в пикселах на дюйм — *ppi*). От установленных значений зависят размер и качество изображения. Для целей электронной публикации лучше установить разрешение 72 *ppi*, для последующей распечатки выбирают разрешение исходя из формулы $ppi \times 1,5 lpi$, где *lpi* — линиятура раstra, установленная на устройстве вывода. Для устройств, поддерживающих стохастическое ЧМ-растрирование, вместо *lpi* надо знать их разрешающую способность в *dpi*. Размер изображения лучше устанавливать в масштабе 1:1 по отношению к тому, что будет использовано в публикации, или несколько больше.

Панель **Инструменты** (рис. 15.18) является одним из основных средств для работы с изображениями. Большинство инструментов, представленных на панели, имеют альтернативные варианты. Их значки помечены маленьким треугольником (разворачивающая кнопка). Если при нажатой кнопке мыши задержать указатель на таком значке, откроется линейка значков с вариантами инструмента.

Панель инструментов разделена на области, в которых сгруппированы средства для редактирования определенных свойств изображения и элементы управления некоторыми параметрами программы.

Первая группа инструментов предназначена для работы с объектами. В нее входят следующие основные инструменты: **Область**, **Лассо**, **Быстрое выделение**,

Волшебная палочка, инструмент Рамка, Раскройка, Выделение фрагмента, Пипетка, Цветовой эталон, Линейка, Комментарий, Счетчик.

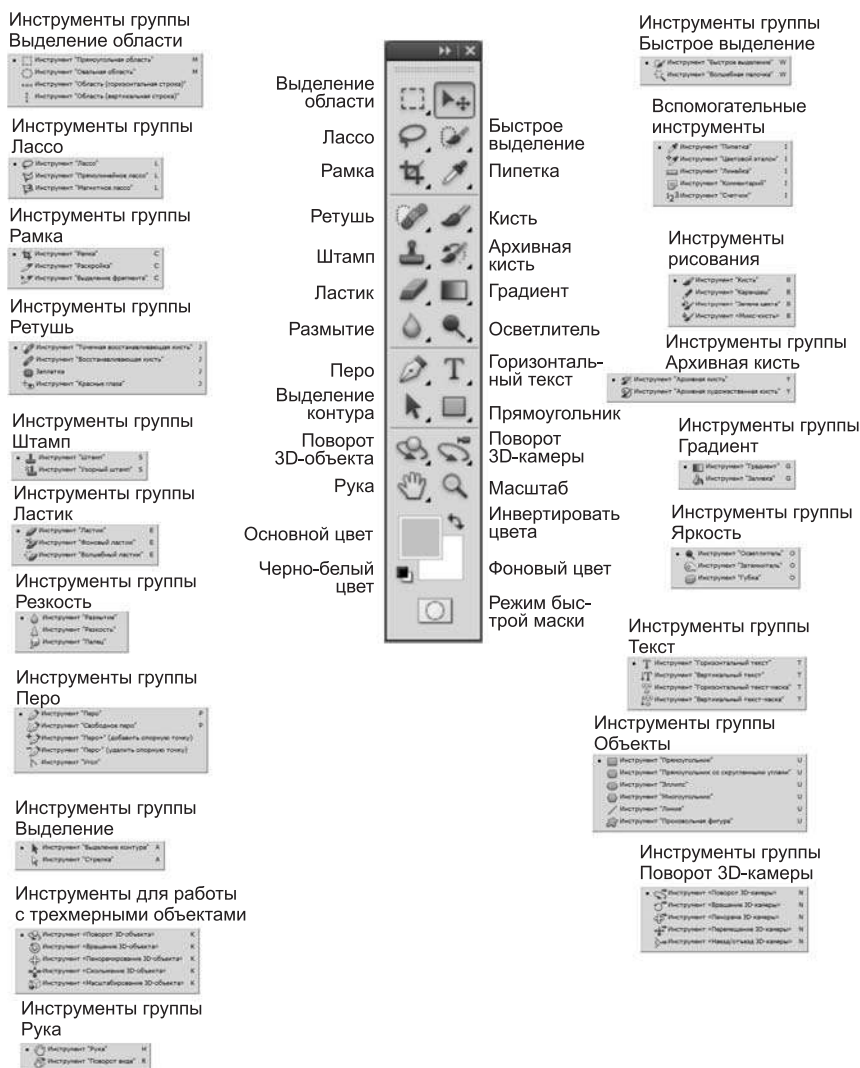


Рис. 15.18. Панель инструментов Adobe Photoshop CS5

Инструментами Область и Лассо выделяют участок изображения, ограниченный геометрической фигурой. Инструмент Волшебная палочка осуществляет выборку области по принципу цветового (или ахроматического) совпадения в рамках границ охвата, установленных пользователем. Инструмент Быстрое выделение позволяет быстро выделить часть рисунка, ориентируясь на область, заданную пользователем. Эти инструменты применяют для выполнения операций *обтравки* — обводки контуров объектов на изображении. Инструментом Перемещение передвигают

выделенные области и копируют их. Инструмент **Рамка** позволяет удалить области (поля) рисунка, а инструмент **Раскройка** используется, когда большой рисунок готовится для отображения в Интернете. С помощью этого инструмента его можно автоматически разделить на фрагменты приемлемого размера, каждый из которых будет загружаться через сеть по отдельности.

Инструмент **Пипетка** служит для точного определения цвета в любой точке изображения и принятия его как образца. Инструмент **Цветовой эталон** позволяет задавать на изображении точки, информацию о цветовых составляющих которых можно отслеживать, пользуясь палитрой **Инфо**. В этом же наборе размещен инструмент **Измерительная линейка**. Инструмент **Комментарий** позволяет привязать к рисунку текстовые замечания. Инструмент **Счетчик** позволяет нумеровать части рисунка.

Следующая группа инструментов предназначена для рисования и ретуши. Она включает инструменты **Восстанавливающая кисть**, **Точечная восстанавливающая кисть**, **Красные глаза**, **Заплата**, **Кисть**, **Карандаш**, **Замена цвета**, **Микс-кисть**, **Штамп** и **Узорный штамп**, **Архивная кисть** и **Архивная художественная кисть**, **Ластик**, **Фоновый ластик** и **Волшебный ластик**, **Градиент** и **Заливка**. Для ретуши также используют инструменты манипуляций с резкостью **Размытие**, **Резкость**, **Палец** или яркостью **Осветлитель**, **Затемнитель**, **Губка**.

Инструмент **Штамп** позволяет выполнять *набивку* — копирование выбранных участков изображения по каждому щелчку мыши. **Узорный штамп** в качестве наполнителя набивки использует узоры, которые можно настроить в панели свойств этого инструмента. **Восстанавливающая кисть** похожа по способу действия на **Штамп**, однако приближает яркость и цвет копируемой области к параметрам пикселей целевой области. Инструмент **Заплата** позволяет выбрать произвольную область-источник протягиванием мыши, а затем переместить ее в нужное место. При этом цвета и яркость области-источника приводятся в соответствие с параметрами пикселей целевой области. **Архивная кисть** приводит параметры пикселей в области действия к исходному состоянию, которое они имели до редактирования. При работе с **Архивной кистью** используется палитра **История**, с помощью которой можно задавать конкретное состояние редактирования изображения для восстановления до него участков изображения, обработанных **Архивной кистью**.

С помощью инструмента **Заливка** выполняют заполнение выделенных участков изображения одним цветом. Плавный переход между цветами обеспечивает инструмент **Градиент**. Инструменты с альтернативным выбором **Резкость/Размытие** позволяют изменять эти параметры на отдельных участках изображения, а инструменты **Осветлитель/Затемнитель/Губка** служат для местной коррекции яркости и цветовой насыщенности.

Третья группа инструментов предназначена для создания новых объектов и редактирования существующих. Инструмент **Перо** позволяет рисовать плавные криволинейные контуры. Инструментом **Текст** выполняют надписи. Инструменты для создания фигур и линий служат для рисования векторных объектов. Для работы с векторными объектами служат инструменты **Выделение контура** и **Стрелка**.

Следующая группа инструментов объединяет средства для работы с трехмерными объектами, а также ряд вспомогательных инструментов. В частности, это инструмент **Рука**, с помощью которого перемещают видимую область по изобра-

жению, и инструмент **Масштаб**, предназначенный для увеличения/уменьшения изображения в видимой области.

В нижней части инструментальной панели помещены средства для работы с цветом и масками. Средство управления цветом показывает основные цвета фона и переднего плана. В левом нижнем углу расположен значок, щелчок на котором устанавливает цвета, принятые по умолчанию для фона и переднего плана. Элемент управления **Редактирование в режиме «Быстрая маска»** позволяет переключаться между стандартным режимом и режимом быстрой маски.

Важнейшим средством управления параметрами инструментов является интерактивная **Панель свойств**. При выборе любого инструмента автоматически появляется соответствующая его свойствам палитра с необходимыми элементами управления.

Помимо **Панели свойств**, многими параметрами инструментов или изображения в целом «заведуют» *инструментальные палитры*, которые представляют собой диалоговые окна особого вида. Всего в программе Adobe Photoshop более двадцати инструментальных палитр.

Управление отображением палитр осуществляется из меню **Окно** с помощью флажков меню, соответствующих существующим палитрам. Неиспользуемые палитры можно также удалить с экрана командой **Заккрыть** контекстного меню палитры. Щелчком на сворачивающей кнопке размер палитры сокращают до размера пиктограммы, которая, в зависимости от ширины настраиваемой панели свернутой палитры, может снабжаться заголовком.

Некоторые палитры имеют командные кнопки, раскрывающиеся списки, поля ввода и другие элементы управления. Назначение конкретного элемента управления поясняет всплывающая подсказка, появляющаяся при задержке указателя мыши на интересующем элементе.

Палитры можно перемещать по экрану методом перетаскивания. Новые палитры создают «сборкой» из имеющихся элементов. Для этого, подцепив указателем мыши корешок одной из вкладок палитры, его перетаскивают в окно другой палитры. Если вкладку разместить на свободном поле экрана, она преобразуется в независимую палитру.

Палитра **3D** служит для работы с трехмерными объектами. Несмотря на то что Adobe Photoshop используется, преимущественно, для работы с двумерной растровой графикой, он содержит элементы векторных и 3D-редакторов.

Палитры **Абзац** и **Символ** применяются для форматирования текстов, вводимых с помощью инструментов работы с текстами.

Палитра **Анимация** применяется при создании анимированных изображений.

Палитры **Гистограмма** позволяет оценить график распределения яркостей пикселей изображения по цветовым каналам. Эту палитру полезно использовать для оценки общего состояния изображения.

Палитра **Журнал измерений** применяется совместно с инструментом **Линейка**. Каждое измерение характеризуется набором параметром, которые и фиксируются в Журнале.

Палитра **Инфо** обеспечивает информационную поддержку средств отображения. На ней представлены: текущие координаты указателя мыши, размер текущей выделенной области, цветовые параметры элемента изображения и другие данные.

Палитра История позволяет перемещаться по истории изменения изображения, с ее помощью осуществляется выбор одного из предыдущих состояний изображения для Архивной кисти.

Палитра Источник клонов предназначена для использования с инструментом Штамп.

Палитру Каналы используют для выделения, создания, дублирования и удаления цветовых каналов, определения их параметров, изменения порядка, преобразования каналов в самостоятельные объекты и формирования совмещенных изображений из нескольких каналов.

Палитра Кисть управляет настройкой параметров инструментов редактирования. Она имеет увеличенный размер, из ее окна можно вызвать вспомогательную палитру Наборы кистей, которая предназначена для быстрого выбора вида и основных параметров кисти.

Палитра Комментарии служит для работы с комментариями, которые можно добавлять к рисунку, используя соответствующий инструмент.

Палитра Композиции слоев позволяет хранить в одном Photoshop-документе несколько вариантов расположения и параметров слоев готового изображения.

Палитра Контуры содержит список всех созданных контуров. Контуры можно использовать для обтравки — программа Adobe Photoshop позволяет преобразовать контур в границу выделенной области.

Палитра Коррекция дает доступ к инструментам коррекции изображения. В частности, с помощью средств этой палитры можно настроить такие параметры, как яркость и контрастность изображения, провести цветокоррекцию и т. д. Палитра, помимо инструментов, имеет набор предустановок.

Палитра Маски позволяет управлять параметрами масок.

Палитра Наборы параметров для инструментов позволяет создавать и хранить настройки различных инструментов.

Палитра Навигатор позволяет просмотреть различные фрагменты изображения и изменить масштаб просмотра. В окне палитры помещена миниатюра изображения с выделенной областью просмотра.

Палитра Образцы содержит набор доступных цветов. Такой набор можно загрузить и отредактировать, добавляя и удаляя цвета. Цветовой тон переднего плана и фона выбирают из состава набора. В стандартном комплекте поставки программы предусмотрено несколько цветовых наборов, в основном компании *Pantone*.

Палитра Операции позволяет создавать *макрокоманды* — заданную последовательность операций с изображением. Макрокоманды можно записывать, выполнять, редактировать, удалять, сохранять в виде файлов.

Палитра Слои служит для управления отображением всех слоев изображения, начиная с самого верхнего. Возможно определение параметров слоев, изменение их порядка, операции со слоями с применением разных методов.

Палитра Стили содержит наборы стилей, которые можно применять к слоям.

Палитра Цвет отображает цветовые значения текущих цветов переднего плана и фона. Ползунки на цветовой линейке соответствующей цветовой системы позволяют редактировать эти параметры.

Особую группу программных средств обработки изображений представляют *фильтры*. Это подключаемые к программе модули, часто третьих фирм, позволяю-

щие обрабатывать изображение по заданному алгоритму. Иногда такие алгоритмы бывают очень сложными, а окно фильтра может иметь множество настраиваемых параметров. Из групп фильтров популярны продукты серий Kai's Power Tools, Alien Skin, DreamSuite и др.

Кроме того, программа содержит набор встроенных фильтров, которые с успехом используются для работы с изображениями.

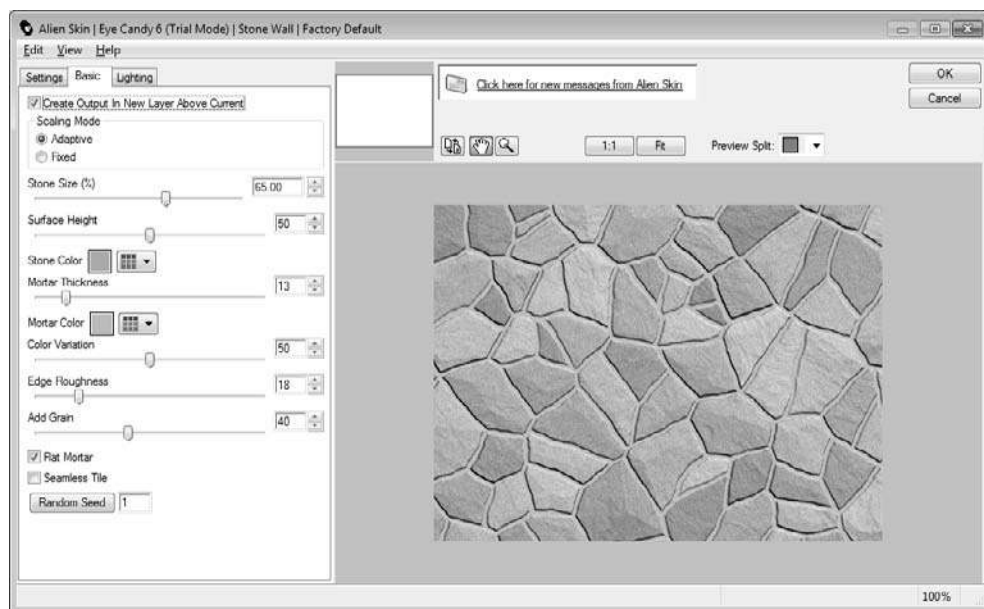


Рис. 15.19. Фильтр Stone Wall из пакета Alien Skin Eye Candy 6 предназначен для создания текстур, имитирующих каменную кладку

15.4. Средства для работы с векторной графикой

Средства создания и обработки векторной графики

К программным средствам создания и обработки векторной графики относятся графические редакторы (например, Adobe Illustrator, CorelDraw) и векторизаторы (трассировщики) — специализированные пакеты преобразования растровых изображений в векторные (например, RxRaster, Easy Trace).

Векторный редактор Adobe Illustrator является одним из общепризнанных лидеров среди программ этого класса. Его особое преимущество заключается в хорошо отлаженном взаимодействии с другими продуктами компании Adobe, прежде всего с пакетами Photoshop, InDesign. Эти приложения выполнены в едином стиле и образуют законченный пакет. Adobe Illustrator имеет встроенные средства по векторизации изображений.

Векторный редактор CorelDraw исторически, особенно в России, считается основным пакетом создания и обработки векторной графики на платформе Windows. К его преимуществам относятся развитая система управления и обширные средства настройки параметров инструментов. По возможностям создания самых сложных

художественных композиций CorelDraw превосходит конкурентов. Однако интерфейс программы считается непростым для освоения.

Основные понятия векторной графики

Основным объектом векторной графики является *линия*. При этом *прямая* линия рассматривается как частный случай *кривой*. Иногда вместо понятия линии используется понятие *контура*. Этот термин более полно отражает суть, поскольку контур может иметь любую форму — прямой, кривой, ломаной линии, фигуры.

Каждый контур имеет две или более *опорных точек*, также именуемых *узлами*. Элемент контура, заключенный между двумя смежными опорными точками, называют *сегментом контура*. *Форму контура* меняют перемещением опорных точек, изменением их свойств, добавлением новых и удалением имеющихся узлов. Контур может быть *открытым* или *замкнутым* — когда последняя опорная точка одновременно является и первой. Свойства замкнутых и открытых контуров различны.

Контур является элементарным графическим *объектом*. Из контуров создают новые объекты или их группы. С несколькими контурами выполняют операции *группировки*, *комбинирования*, *объединения*. В результате образуются соответственно: *группа объектов*, *составной контур*, *новый контур*. После операции группировки каждый контур сохраняет свои свойства и принадлежащие ему узлы. После операции комбинирования составной контур приобретает новые свойства, но узлы остаются прежними. После операции объединения образуются новые узлы и меняются свойства исходных контуров (рис. 15.20).

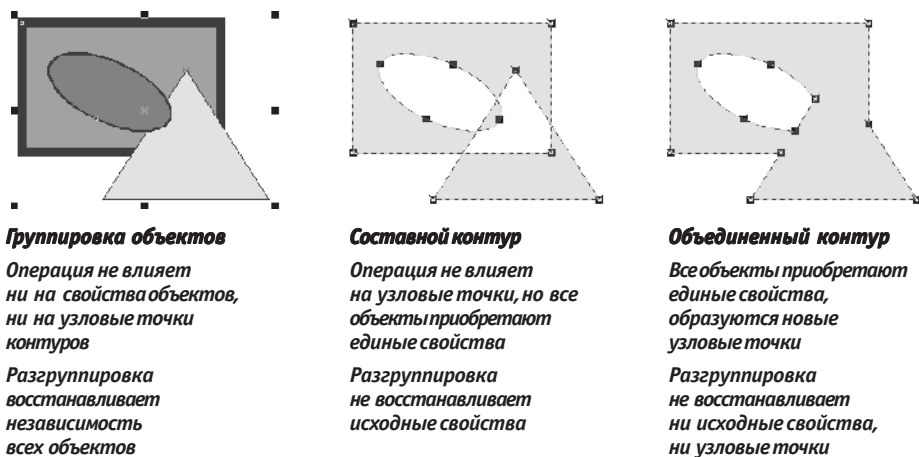


Рис. 15.20. Действия с группой векторных объектов

Параметры обводки контура определяют его вид при отображении. К ним относятся:

- ☐ толщина линии;
- ☐ цвет линии;
- ☐ тип линии (сплошная, пунктирная и пр.);
- ☐ форма концов (со стрелкой, закругленные и пр.).

Замкнутые контуры обладают особым свойством — *заливкой*, то есть параметрами заполнения охватываемой области. Заливка также является объектом и обладает собственным набором свойств. Различают несколько типов заливки:

- ☐ заливка основным цветом, то есть заполнение внутренней области избранным цветом;
- ☐ градиентная заливка — заполнение двумя или более цветами с плавным переходом между ними;
- ☐ текстурная заливка — заполнение узором с регулярной структурой;
- ☐ заливка изображением-картой — заполнение готовым растровым изображением, называемым *картой*;
- ☐ заливка векторным изображением (текстуры PostScript).

Векторный редактор Adobe Illustrator

Редактор Adobe Illustrator удобен для изучения начинающими пользователями по причине наличия официальных локализованных версий, сравнительно простого интерфейса и развитых функциональных возможностей (рис. 15.21).

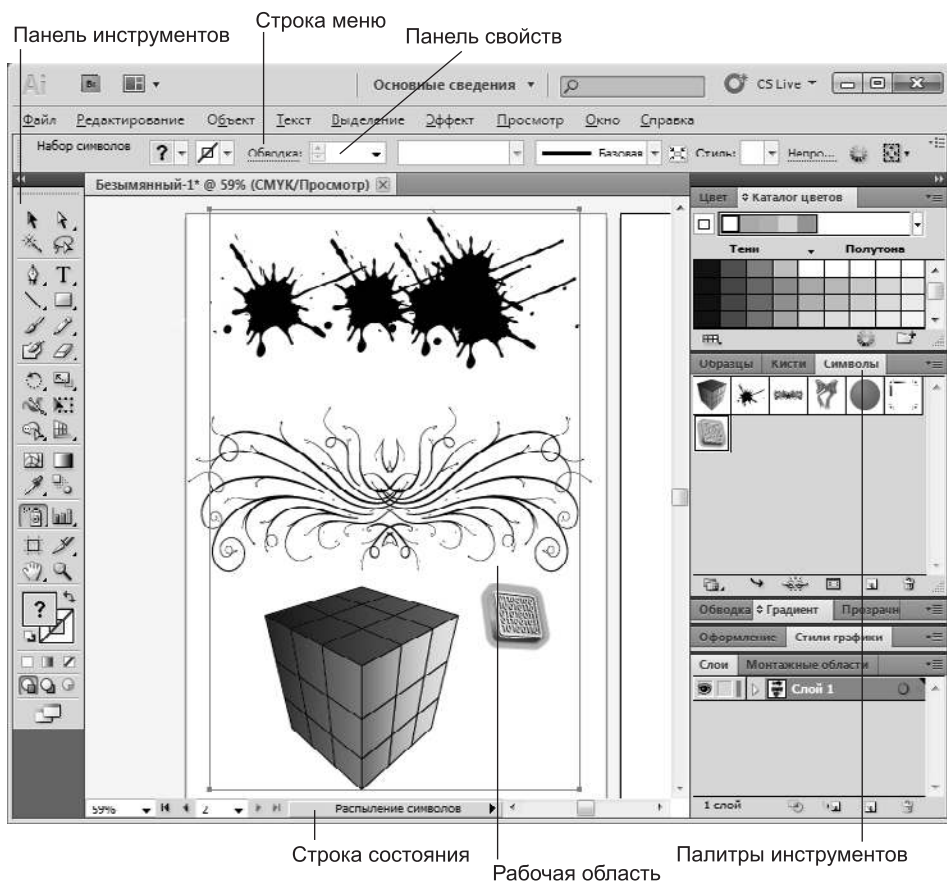


Рис. 15.21. Рабочее окно векторного редактора Adobe Illustrator

Основные элементы управления программы Adobe Illustrator сосредоточены в строке меню, на панели инструментов и в инструментальных палитрах.

Панель инструментов выполнена подобно рассмотренной нами ранее для Adobe Photoshop и включает девять групп значков (рис. 15.22).

Первая группа объединяет инструменты выделения объектов. Инструмент **Выделение** позволяет выделить объект целиком щелчком на его контуре или построением рамки вокруг объекта. Инструмент **Прямое выделение** служит для выделения части контура, например одного сегмента. При нажатой клавише **SHIFT** этими инструментами выделяют несколько объектов.

Инструментом **Лассо** выделяют объекты, хотя бы частично попавшие внутрь очерченной указателем области. Инструмент **Волшебная палочка** позволяет выделять объекты с одинаковыми свойствами контура и заливки.

Для рисования предназначены инструменты второй группы, объединяющие средства создания фигур, линий и текста. Инструмент **Перо** и его альтернативные варианты используют для рисования кривых и редактирования их точек. Назначение инструмента **Текст** понятно из его названия. Альтернативные варианты инструмента позволяют разместить текст в окне (текстовом блоке), по вертикали, под наклоном и вдоль кривой. Линии и сегменты заданной формы рисуют с помощью инструментов **Отрезок линии**, **Дуга**, **Спираль**, **Прямоугольная сетка**, **Полярная сетка**. Для этих инструментов можно заранее задать свойства создаваемого объекта в диалоговом окне, открываемом при двойном щелчке на значке инструмента или одинарном щелчке на рабочем поле.

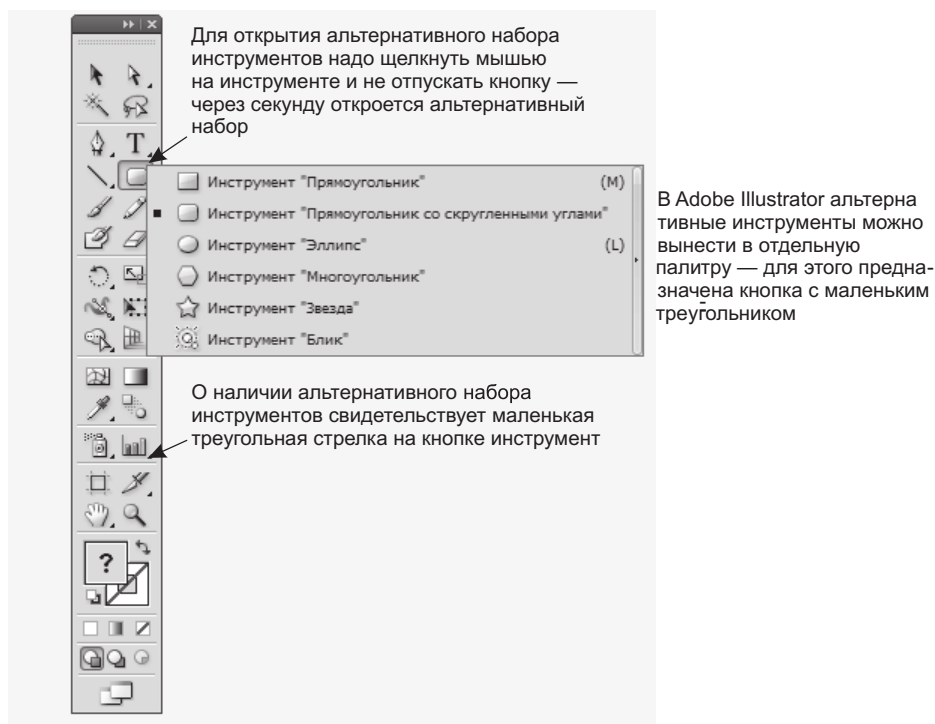


Рис. 15.22. Панель инструментов Adobe Illustrator

Инструменты Прямоугольник, Прямоугольник со скругленными углами, Эллипс, Многоугольник, Звезда, Блик служат для создания соответствующих геометрических фигур. Инструмент Кисть расставляет вдоль рисуемого контура объекты, образующие Узор или Каллиграфическую линию. Инструментом Карандаш рисуют контуры произвольной формы (с автоматической расстановкой узлов). Альтернативными вариантами карандаша служат инструменты Сглаживание и Стирание контура. Первый позволяет улучшить гладкость контура, а второй удаляет опорные точки контура в области своего воздействия. Инструмент Ластик предназначен для стирания графических объектов, инструменты Ножницы и Нож предназначены для разделения контуров и разрезания объектов.

Инструменты третьей группы предназначены для различных преобразований объектов. Суть этих операций ясна из названий инструментов: Поворот, Зеркальное отражение, Масштаб, Наклон, Перерисовка, Ширина, Деформация, Воронка, Втягивание, Раздувание, Зубцы, Кристаллизация, Морщины, Свободное трансформирование, Создание фигур, Быстрая заливка, Выделение быстрых заливок, Сетка перспективы, Выбор перспективы.

В следующую группу входят инструменты: Сетчатый градиент, Градиент, Пипетка, Линейка и Переход.

Пятая группа содержит инструмент Symbol Sprayer (Распылитель символов) и его альтернативные варианты, а также средства построения диаграмм (Вертикальные полосы и др.) различной формы.

В следующей группе объединены инструменты вспомогательного назначения: Монтажная область, Фрагменты, Выделение фрагмента, Рука, Разбиение для печати и Масштаб.

Элементы управления в нижней части Панели инструментов практически совпадают с рассмотренными выше для программы Adobe Photoshop. Среди отличий можно отметить наличие кнопок выбора способа заливки создаваемых объектов (Цвет, Градиент, Без цвета), выбор режима рисования (Рисовать в стандартном режиме, Рисовать за объектами, Рисовать внутри объектов). Нижнюю часть Панели инструментов занимает кнопка для быстрого переключения между экранными режимами программы.

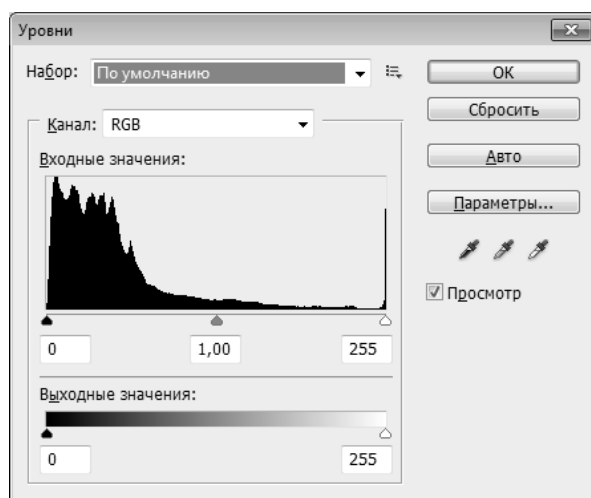
В векторном редакторе Adobe Illustrator также применяются *палитры инструментов*. Их общие свойства и методы управления отображением, настройки параметров совпадают с таковыми в редакторе Adobe Photoshop. Всего в редакторе Adobe Illustrator CS 5 предусмотрено более 25 палитр. К сожалению, параметры и свойства объекта разнесены по многим палитрам, и потому настройка нужных свойств вызывает определенные сложности у начинающих пользователей.

В целом графический редактор Adobe Illustrator обладает обширными возможностями и достаточно прост в освоении.

Практическое занятие

Упражнение 15.4. Изменение динамического диапазона изображения

1. Запустите программу Adobe Photoshop, откройте файл, находящийся по адресу C:\Program Files\Adobe\Adobe Photoshop CS5\Образцы\Наложить на HDR\2.jpg, или файл, указанный преподавателем.
2. Оцените динамический диапазон изображения, то есть разброс между минимальной и максимальной яркостью. Для этого откройте окно Изображения ► Коррекция ► Уровни. По гистограмме снимка видно, что диапазон яркостей сдвинут в область темных полутонов, то есть изображение не охватывает доступный динамический диапазон.



3. По гистограмме видно, что область темных тонов значительно превосходит область светлых тонов — снимок выглядит темным. Для исправления изображения выполните *гамма-коррекцию*. Откройте диалоговое окно Изображения ► Коррекция ► Кривые. Указателем мыши перетащите середину гамма-кривой вверх, «подтягивая» полутона к более светлым. При этом окончания кривой (то есть границы динамического диапазона) остаются неизменными. Регулируя кривизну линии, добейтесь наиболее сбалансированной яркости и контрастности. Зафиксируйте результат щелчком на кнопке OK.
4. Испытаем другой способ гамма-коррекции — с помощью диалогового окна Уровни. В среднем окне поля Входные значения установим гамма-коэффициент, больший единицы. Или переместим средний движок влево, отслеживая изменения гамма-коэффициента. Таким способом добиваемся повышения качества изображения.
5. Некоторые изображения имеют тусклый, малоконтрастный вид. Если проанализировать гистограмму распределения яркостей таких изображений

в окне **Уровни**, можно заметить, что гистограмма имеет ярко выраженный «холм» в середине диапазона. Для того чтобы расширить динамический диапазон изображения и улучшить его внешний вид, можно переместить правый и левый бегунки, находящиеся под гистограммой, ближе к центру, добившись экспериментальным путем приемлемого результата, или ввести в правое и левое поля значения больше 0 и меньше 255 соответственно.

Итог. Мы установили, что управление динамическим диапазоном изображения позволяет существенно улучшить качество растровой графики. Мы также выяснили, что существует зависимость между яркостью и контрастностью изображения.

Упражнение 15.5. Ретушь изображения

1. Запустите программу Adobe Photoshop, откройте файл, указанный преподавателем.

Для правильного выполнения данного упражнения следует подготовить отсканированную копию старого, малоконтрастного, поврежденного фотоснимка.

2. Найдите и выделите слишком темные участки снимка. Выберите инструмент **Лассо**, на панели свойств установите флажок **Сглаживание** и задайте значение растушевки (**Растушевка**) 4 пиксела. Инструментом **Лассо** выделите темные области. Для последовательного выделения нескольких несвязанных областей после выделения первой области переключите **Лассо** в режим **Добавить к выделенной области** с помощью соответствующей кнопки на панели свойств инструмента.
3. Для осветления выделенной области откройте диалоговое окно **Изображения ► Коррекция ► Уровни** и, перемещая движки, добейтесь такой яркости и контрастности изображения, чтобы тень исчезла.
4. Удалите повреждения на изображении. Для этого воспользуйтесь инструментами **Штамп** и **Восстанавливающая кисть**. Процесс заключается в забивке ненужной детали переносом фрагментов соседних областей.

В палитре **Слои** щелкните правой кнопкой мыши на слое **Фон** и в открывшемся меню выберите пункт **Создать дубликат слоя**. Щелчком на новом слое сделайте его активным.

Выберите инструмент **Штамп**. На панели свойств в разделе **Кисть** щелкните на раскрывающейся кнопке, в открывшемся диалоговом окне установите диаметр кисти движком **Размер**, движок **Жесткость** установите в значение 15%.

Установите указатель мыши на исходный участок изображения и, удерживая клавишу **ALT**, щелкните мышью, фиксируя зону, с которой будет происходить перенос изображения. Исходная точка в момент щелчка отмечается крестообразным маркером. Переместите указатель мыши на ближайшую область фона и щелчками закрасьте ее. При необходимости изменяйте исходную точку щелчком с нажатой клавишей **ALT**. С помощью инструмента **Штамп** удалите примерно половину повреждений.

5. Действуя аналогичным образом, с помощью инструмента **Восстанавливающая кисть** удалите остальные повреждения на снимке.
6. Выполните местную коррекцию резкости. Выберите инструмент **Резкость**. На панели свойств выберите подходящий размер области действия инструмента. Проведите улучшение резкости лица для усиления рельефности деталей — изображение станет выразительнее.



Рис. 15.23. Рисунок до ретуши (слева) и после нее (справа)

Итог. Инструменты местной коррекции и ретуши изображения позволяют восстановить поврежденные и старые фотоснимки, улучшить восприятие деталей изображения, убрать ненужные детали, подчеркнуть важные элементы изображения.

Упражнение 15.6. Использование фильтров

1. Запустите программу Adobe Photoshop, откройте файл, указанный преподавателем.

Для правильного выполнения данного упражнения следует подготовить отсканированную копию старого, малоконтрастного, поврежденного фотоснимка.

Удаляем пыль и царапины со снимка. Дадим команду **Фильтр ▶ Шум ▶ Пыль и царапины**. Обратите внимание, что фильтр сглаживает границы элементов изображения. Он удобен при ретуши старых, поврежденных снимков и воздействует на всю поверхность изображения.

2. Повышаем резкость изображения. Дадим команду **Фильтр ▶ Усиление резкости ▶ Резкость на краях**. Обратите внимание, что фильтр воздействует только на границы перехода между элементами изображения разной яркости.

3. Проведите эксперименты с фильтрами группы **Стилизация**.
4. Проведите эксперименты с фильтрами группы **Пикселизация**.

Итог. Мы освоили приемы применения фильтров для улучшения качества изображения за счет ретуши и акцента на важных деталях. Возможна имитация размещения изображений на разных материалах, изменение условий освещенности и другие художественные эффекты. Применять фильтры надо осторожно, чтобы не испортить, а усилить впечатление, которое должно вызывать изображение.

Упражнение 15.7. Обтравка изображения

1. Запустите программу Adobe Photoshop, откройте файл, указанный преподавателем.

Для правильного выполнения данного упражнения следует подготовить снимок, на котором изображен какой-либо объект, подходящий для экспериментов по обтравке.

2. Процесс точного выделения элемента изображения называют *обтравкой*. Выберите инструмент **Магнитное лассо**. На панели свойств установите нулевую величину **Растушевка**.
3. Обведите как можно точнее инструментом **Магнитное лассо** контур фигуры. Начните работу с установки первой ключевой точки на границе объекта. Проведите инструмент вдоль границы изображения. Линия автоматически «пристанет» к границе. Поставьте следующую ключевую точку для того, чтобы закрепить результат выделения участка контура, и продолжите работу до тех пор, пока не вернетесь в исходную точку. Выделение завершается щелчком в исходной ключевой точке.
4. Испытайте альтернативный способ выделения области на изображении. Выберите инструмент **Волшебная палочка**. Установите на панели свойств в поле **Чувствительность** значение 24. Щелкая инструментом, выделите фон вокруг фигуры. Для добавления или исключения областей выделения удерживайте в момент щелчка клавиши **SHIFT** или **ALT**.
5. Выполните окончательную корректировку выделенной области с помощью инструмента **Лассо**.
6. Обтравочный контур может храниться в том же файле, но отдельно от изображения. Для этого предназначены каналы. В палитре **Каналы** щелкните на кнопке сохранения выделенной области в новом канале. Задайте новому каналу имя. В окне палитры появляется миниатюра с изображением обтравочного контура. Загрузку контура из канала производят щелчком на его изображении при нажатой клавише **CTRL**.

Итог. Каналы и обтравочные контуры служат мощным средством композиции и редактирования изображений. Их умелое применение позволяет создавать качественную рекламную и художественную продукцию, в которой использована вся сила инструментария Adobe Photoshop.

Исследовательская работа

Задание 15.1. Составление композиции в программе Adobe Photoshop

1. Запустите программу Adobe Photoshop, откройте файлы C:\Program Files\Adobe\Adobe Photoshop CS5\Образцы\Photomerge\Фото2.jpg и C:\Program Files\Adobe\Adobe Photoshop CS5\Образцы\Утенок.tif.
2. Улучшите динамический диапазон изображения Фото2.jpg командой Изображение ► Коррекция ► Уровни ► Авто.
3. Выберите инструмент Магнитное лассо, в файле Утенок.tif выполните обтравку фигуры.
4. Создайте канал для контура обтравки в изображении Утенок.tif. Для этого щелкните в палитре Каналы на кнопке сохранения контура в новом канале.
5. Улучшите динамический диапазон выделенной фигуры командой Изображение ► Коррекция ► Уровни ► Авто.
6. Выполните гамма-коррекцию выделенного изображения, задав коэффициент 0,85 в окне Изображение ► Коррекция ► Уровни.
Примените растушевку краев области выделения, установив в поле Растушевка значение 4px.
7. Скопируйте выделенную область в буфер обмена командой Редактирование ► Скопировать.
8. Вклейте скопированную область на новый слой изображения Фото2.tif командой Редактирование ► Вставить.
9. Установите на палитре Слои уровень непрозрачности нового слоя 50%. Получится изображение, представленное на рис. 15.24.
10. Поэкспериментируйте с режимом совмещения слоев с помощью раскрывающегося списка в палитре Слои.
11. Попробуйте действие различных фильтров, применяя их отдельно к слоям. Запишите названия и параметры примененных фильтров.

Слой	Название фильтра	Параметры фильтра

Итог. Мы выяснили, что программа Adobe Photoshop позволяет достаточно легко создавать сложные композиции с применением специальных эффектов.



Рис. 15.24. Композиция, выполненная с использованием дополнительного слоя

Практическое занятие

Упражнение 15.8. Создание простейших объектов в редакторе Adobe Illustrator

1. Запустите векторный редактор Adobe Illustrator.
2. Создайте новый документ командой Файл ► Новый.
3. Установите параметры страницы в диалоговом окне: в поле Размер — значение А4, в поле Единицы измерения — значение Миллиметры, переключателем Ориентация — книжную ориентацию.
4. *Рисование прямой линии.* Выберите инструмент Перо. Первым щелчком задайте начальную опорную точку, вторым щелчком после смещения указателя мыши задайте конечную опорную точку. На палитре Обводка установите толщину 8 пунктов. При нажатой клавише CTRL щелкните на свободном поле. Далее создайте строго вертикальную линию. При нажатой клавише SHIFT сделайте первый щелчок, удерживая клавишу, переместите указатель вниз и сделайте второй щелчок.

5. *Рисование замкнутого контура.* Выберите инструмент **Перо**. Щелчком задайте начальную опорную точку, затем сделайте четыре щелчка вдоль воображаемого контура многоугольника. Подведите указатель к начальной опорной точке так, чтобы рядом с его значком появился кружок. В этот момент сделайте последний щелчок.
6. *Рисование эллипса и окружности.* Выберите инструмент **Эллипс**. Щелкните на рабочем поле и протягиванием задайте форму и размеры эллипса. Удерживая клавишу **SHIFT**, вновь щелкните на рабочем поле и протягиванием создайте правильную окружность. Для рисования фигуры «от центра» удерживайте комбинацию клавиш **ALT+SHIFT**, щелкните на рабочем поле и создайте правильную окружность, начиная от ее центра.
7. *Рисование спирали.* Щелкните на значке **Отрезок линии** инструментальной панели и удерживайте кнопку до появления линейки с альтернативным набором инструментов. Выберите инструмент **Спираль**. Щелкните на рабочем поле и протягиванием от центра создайте спираль. Следующую спираль создайте методом задания параметров. Щелкните на рабочем поле и в открывшемся диалоговом окне **Спираль** задайте необходимые параметры.
8. *Рисование многоугольника.* Выберите инструмент **Многоугольник**. Щелкните на рабочем поле и в появившемся окне установите число его вершин равным двенадцати.

Итог. Мы научились создавать простейшие векторные объекты с помощью векторного графического редактора Adobe Illustrator. Мы установили, как с помощью клавиш **SHIFT** и **ALT** можно модифицировать действие чертежных инструментов.

Упражнение 15.9. Создание криволинейных контуров

1. Запустите векторный редактор Adobe Illustrator.
2. Создайте новый документ командой **Файл ► Новый**.
3. Установите параметры страницы в диалоговом окне: в поле **Размер** — значение **A4**, в поле **Единицы измерения** — значение **Миллиметры**, переключателем **Ориентация** — книжную ориентацию.
4. *Рисование контура произвольной формы.* Выберите инструмент **Карандаш**. Удерживая нажатой кнопку мыши, протягивайте указатель по рабочему полю, рисуя волнистую линию. Выберите инструмент **Кисть**. Удерживая нажатой кнопку мыши, протягивайте указатель по рабочему полю, рисуя волнистую линию.
5. *Построение кривых.* Выбираем инструмент **Перо**. Создайте три кривые, основанные на разных математических формулах (рис. 15.25):
 - кривая первого порядка — щелчок + щелчок;
 - кривая второго порядка — щелчок + протягивание + щелчок;
 - кривая третьего порядка — щелчок + протягивание + щелчок + протягивание.

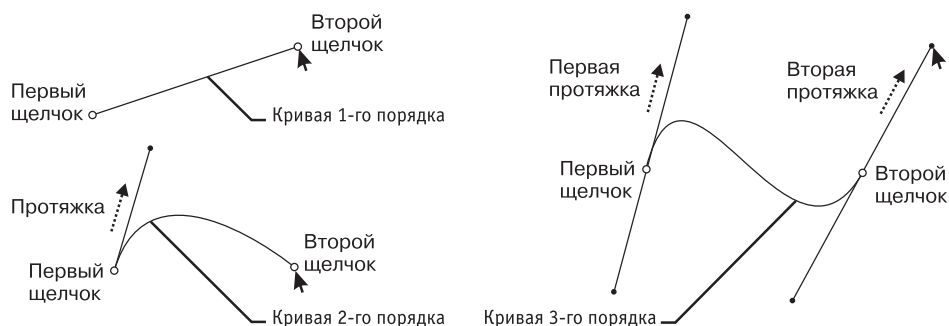


Рис. 15.25. Построение кривых первого, второго и третьего порядка

6. *Создание замкнутого криволинейного контура.* Выберите инструмент **Перо**. Пользуясь приемами, описанными в предыдущем пункте, создайте контур. Последний щелчок выполните, подведя указатель к начальной опорной точке (после появления кружка у его значка).

Упражнение 15.10. Редактирование контуров

1. Запустите векторный редактор Adobe Illustrator.
2. Создайте новый документ командой **Файл ▶ Новый**.
3. Установите параметры страницы в диалоговом окне: в поле **Размер** — значение **A4**, в поле **Единицы измерения** — значение **Миллиметры**, переключателем **Ориентация** — книжную ориентацию.
4. Создайте замкнутый криволинейный контур. Выберите инструмент **Перо**. Щелчками с последующим протягиванием создайте на рабочем поле замкнутый криволинейный контур.
5. *Перемещение опорных точек.* Выберите инструмент **Прямое выделение**. Выделите опорную точку на криволинейном контуре и протягиванием сместите ее так, чтобы изменить форму контура. Повторите операцию для двух-трех опорных точек.
6. *Изменение свойств опорных точек.* Выберите инструмент **Преобразовать опорную точку**. Щелчком на *гладкой* опорной точке контура преобразуйте ее в *угловую*. Щелчком на *гладкой* опорной точке с последующим протягиванием преобразуйте ее в *симметричную*. Установите указатель на *управляющую линию*, изменением ее длины и угла наклона касательной измените форму криволинейного контура.
7. *Создание и удаление опорных точек.* Выберите инструмент **Добавить опорную точку**. Выберите сегмент на криволинейном контуре и щелчком добавьте новую опорную точку. Выберите инструмент **Удалить опорную точку**. Щелчком на опорной точке криволинейного контура удалите ее.

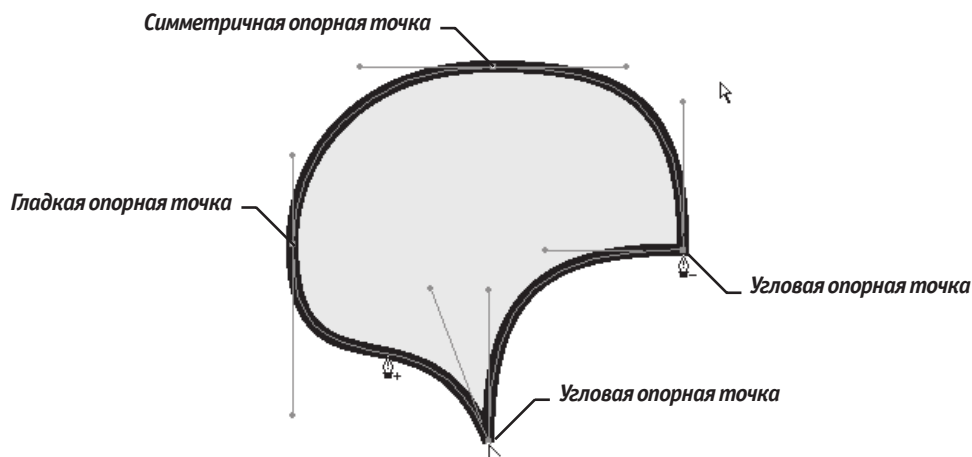


Рис. 15.26. Редактирование контура

Упражнение 15.11. Обработка замкнутых контуров

1. Запустите векторный редактор Adobe Illustrator.
2. Создайте новый документ командой **Файл** ► **Новый**.
3. Установите параметры страницы в диалоговом окне: в поле **Размер** — значение **A4**, в поле **Единицы измерения** — значение **Миллиметры**, переключателем **Ориентация** — книжную ориентацию.
4. Создайте замкнутый криволинейный контур. Выберите инструмент **Перо**. Щелчками с последующим протягиванием создайте на рабочем поле замкнутый криволинейный контур. Откройте палитры **Цвет**, **Градиент**, **Образцы** (**Окно** ► **Имя палитры**).
5. *Выполнение сплошной заливки.* Инструментом **Selection** (**Выделение**) выберите криволинейный контур. На панели инструментов щелкните по кнопке механизма заливки **Цвет**. Выберите цвет заливки щелчком на образце на палитре **Образцы**. Меняйте цвет заливки щелчками на цветовой линейке палитры **Цвет**. Меняйте цвет заливки перемещением движков (**C**, **M**, **Y**, **K**) на палитре **Цвет**.
6. *Выполнение градиентной заливки.* Инструментом **Выделение** выберите криволинейный контур. На панели инструментов щелкните на кнопке механизма заливки **Градиент**. Выберите исходный цвет в палитрах **Образцы** или **Цвет** методами, описанными в предыдущем пункте. На палитре **Градиент** в раскрывающемся списке **Тип** укажите **Линейный**. В поле **Угол** задайте направление **45** градусов. На **градиентной линейке** щелчком выделите **маркер концевого цвета**, затем на палитре **Цвет** определите его цветовой тон. Перемещением маркеров концевых цветов и **маркера срединной точки** задайте необходимую градиентную растяжку.

7. *Выполнение текстурной заливки.* Инструментом **Выделение** выберите криволинейный контур. В палитре **Образцы** щелчком на кнопке **Меню библиотек образцов** откройте комплект образцов. Щелчком на образце назначьте параметры текстурной заливки.

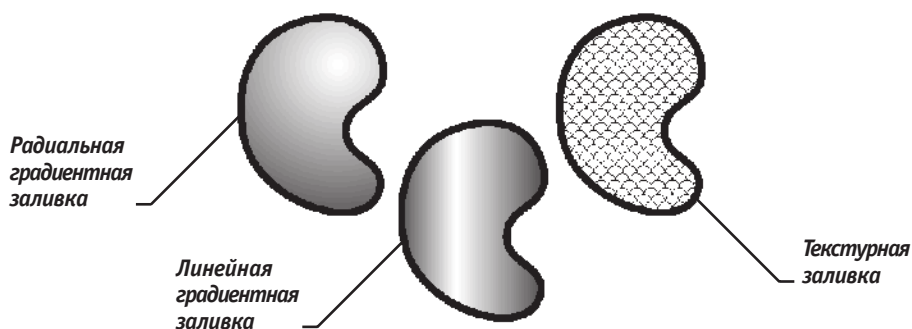


Рис. 15.27. Различные виды заливок замкнутых контуров

8. *Размыкание замкнутого контура.* Выберите инструмент **Ножницы**. Щелчком на сегменте криволинейного контура разомкните его в избранной точке.

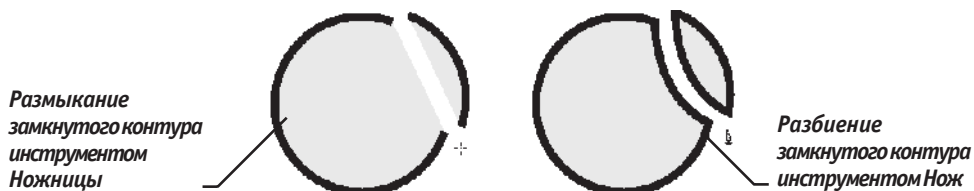


Рис. 15.28. Разделение контуров методом размыкания и разбиения

9. *Разбиение замкнутого контура.* Выберите инструмент **Нож**. При нажатой кнопке мыши протащите указатель поперек криволинейного контура. Выделите один из получившихся объектов инструментом **Прямое выделение** и перетащите в сторону.

Итог. В векторной графике замкнутые контуры обладают особым свойством — заливкой. В этом упражнении мы научились управлять этим свойством средствами редактора Adobe Illustrator. Мы изучили несколько методов создания заливки разных типов и познакомились с приемами размыкания и разбиения замкнутых контуров.

Исследовательская работа

Задание 15.2. Создание сложных композиций средствами Adobe Illustrator

1. Запустите векторный редактор Adobe Illustrator.
2. Создайте новый документ командой **Файл ▶ Новый**.

3. Установите параметры страницы в диалоговом окне: в поле **Размер** — значение **A4**, в поле **Единицы измерения** — значение **Миллиметры**, переключателем **Ориентация** — книжную ориентацию.
4. Выберите инструмент **Перо**. Щелчками с последующим протягиванием создайте на рабочем поле замкнутый криволинейный контур. На палитре **Обводка** установите толщину обводки 10 пунктов.
5. Выберите инструмент **Эллипс**. Щелкните на рабочем поле и протягиванием задайте форму и размеры эллипса. На палитре **Обводка** установите толщину обводки 4 пункта. Удерживая клавишу **SHIFT**, вновь щелкните на рабочем поле и протягиванием создайте правильную окружность. На палитре **Обводка** установите толщину обводки 7 пунктов.
6. Откройте палитры **Цвет**, **Градиент**, **Образцы** (**Окно** ► **Имя палитры**).
7. Инструментом **Выделение** выберите окружность. На панели инструментов щелкните по кнопке механизма заливки **Градиент**. Выберите исходный цвет в палитре **Образцы**. На палитре **Градиент** в раскрывающемся списке **Тип** укажите **Линейный**. В поле **Угол** задайте направление 0 градусов. На градиентной линейке щелчком выделите маркер концевой цвета, затем на палитре **Цвет** определите его цветовой тон. Перемещением маркеров концевых цветов маркера срединной точки задайте необходимую градиентную растяжку.
8. Инструментом **Выделение** выберите эллипс. На палитре **Образцы** щелчком на кнопке **Меню библиотек образцов** откройте комплект образцов. Щелчком на образце назначьте параметры текстурной заливки.
9. Инструментом **Выделение** выберите криволинейный контур. На панели инструментов щелкните на кнопке механизма заливки **Цвет**. Выберите цвет заливки щелчком на образце в палитре **Образцы**.
10. *Группировка объектов.* Инструментом **Выделение** при нажатой клавише **SHIFT** выберите все объекты. Дайте команду **Объект** ► **Сгруппировать**. Запишите, изменились ли свойства объектов:
 - а) обводка _____ ;
 - б) заливка _____ .
11. *Разгруппировка.* Инструментом **Выделение** выберите сгруппированные объекты. Дайте команду **Объект** ► **Разгруппировать**.
12. *Объединение контуров.* Инструментом **Выделение** выберите окружность и перетащите ее до частичного наложения на эллипс. Окружность должна находиться сверху эллипса, так как создавалась последней. При нажатой клавише **SHIFT** выберите эллипс. На панели **Обработка контуров** щелкните на кнопке **Соединение**. Запишите, как изменились свойства объектов.

Объект	Обводка, толщина в пунктах		Заливка, тип	
	до операции	после операции	до операции	после операции
Окружность				
Эллипс				
Результирующий				

13. *Пересечение контуров.* Отмените предыдущую операцию командой Редактирование ► Отменить. Инструментом Выделение выберите окружность. При нажатой клавише SHIFT выберите эллипс. На панели Обработка контуров щелкните на кнопке Пересечение. Запишите, как изменились свойства объектов.

Объект	Обводка, толщина в пунктах		Заливка, тип	
	до операции	после операции	до операции	после операции
Окружность				
Эллипс				
Результирующий				

14. *Исключение контуров.* Отмените предыдущую операцию командой Редактирование ► Отменить. Инструментом Выделение выберите окружность. При нажатой клавише SHIFT выберите эллипс. На панели Обработка контуров щелкните на кнопке Исключение. Запишите, как изменились свойства объектов.

Объект	Обводка, толщина в пунктах		Заливка, тип	
	до операции	после операции	до операции	после операции
Окружность				
Эллипс				
Результирующий				

15. *Операция Минус верхний.* Отмените предыдущую операцию командой Редактирование ► Отменить. Инструментом Выделение выберите окружность. При нажатой клавише SHIFT выберите эллипс. На панели Обработка контуров щелкните на кнопке Минус верхний. Запишите, как изменились свойства объектов.

Объект	Обводка, толщина в пунктах		Заливка, тип	
	до операции	после операции	до операции	после операции
Окружность				
Эллипс				
Результирующий				

16. *Операция Минус нижний.* Отмените предыдущую операцию командой Редактирование ► Отменить. Инструментом Выделение выберите эллипс. При нажатой клавише SHIFT выберите окружность. На панели Обработка контуров щелкните на кнопке Минус нижний. Запишите, как изменились свойства объектов.

Объект	Обводка, толщина в пунктах		Заливка, тип	
	до операции	после операции	до операции	после операции
Окружность				
Эллипс				
Результирующий				

17. *Комбинирование контуров.* Отмените предыдущую операцию командой Редактирование ► Отменить. Инструментом Выделение выберите окружность. При нажатой клавише SHIFT выберите эллипс. Дайте команду Объект ► Составной контур ► Создать. Запишите, как изменились свойства объектов.

Объект	Обводка, толщина в пунктах		Заливка, тип	
	до операции	после операции	до операции	после операции
Окружность				
Эллипс				
Результирующий				

Глава шестнадцатая

Векторный редактор Corel Draw

16.1. Особенности CorelDraw

Векторный графический редактор CorelDraw канадской фирмы Corel Corporation получил известность благодаря широким возможностям, наличию огромных библиотек готовых изображений, мощной встроенной системе обучения и подсказок. Некоторые средства CorelDraw не имеют аналогов в других векторных редакторах и делают его уникальным продуктом. Тесная интеграция CorelDraw с другим ПО Corel позволяет создать законченную систему подготовки электронных и полиграфических публикаций.

Интерфейс CorelDraw выполнен в традициях, ставших стандартом де-факто для приложений в операционной системе Windows. Окно программы имеет стандартные элементы: рабочее поле, строку заголовка, строку меню, строку состояния, панель инструментов, другие панели, состав которых может определять пользователь, полосы прокрутки и пр. Однако интерфейс CorelDraw имеет и свои особенности, отличающие его от других программ.

Конфигурация интерфейса может быть настроена аналогично другим популярным графическим пакетам. То есть пользователь, привыкший к работе в среде этих редакторов, при переходе к CorelDraw может остаться в рамках уже освоенного интерфейса.

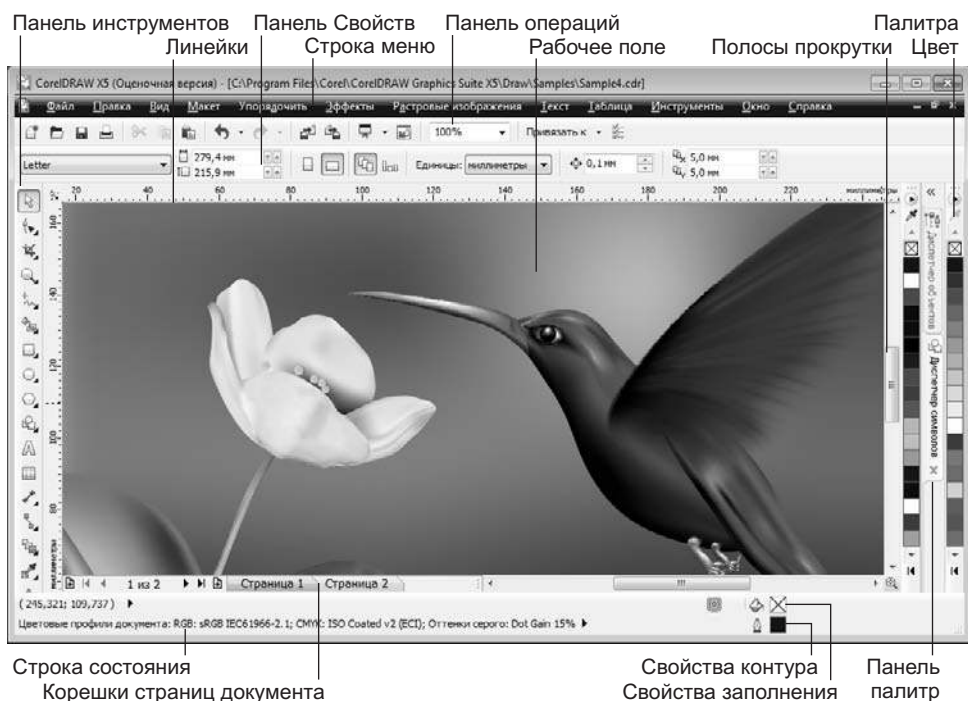


Рис. 16.1. Рабочее окно векторного редактора CorelDraw

16.2. Элементы управления

Панель инструментов

Основные средства создания и редактирования объектов размещены в **Наборе инструментов**. Ряд инструментов имеет альтернативные варианты, что опознается по наличию раскрывающейся кнопки (черный треугольник) в правом нижнем углу значка. Раскрыв панель альтернативного выбора инструментов, ее можно перетащить за корешок на **Рабочее поле**, превратив тем самым в самостоятельную палитру

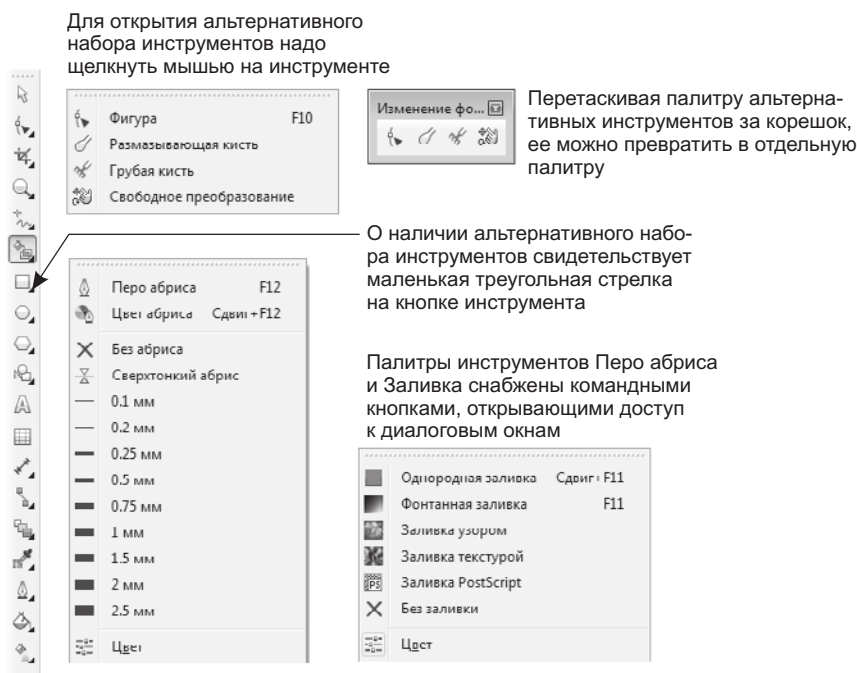
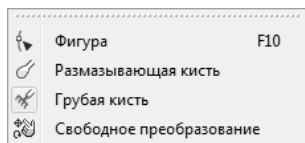


Рис. 16.2. Панель инструментов CorelDraw

Далее приведен полный перечень инструментов CorelDraw X5 (версия 15).



Инструмент **Выбор** позволяет выбирать объекты, изменять их размеры, вращать или искажать их форму.



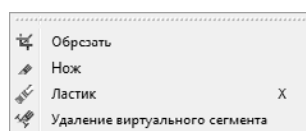
Инструмент **Фигура** предназначен для изменения формы объектов путем выборки опорных точек и редактирования их свойств.

Инструмент Размазывающая кисть позволяет искажать форму контура в соответствии с траекторией

ПЯТНА КИСТИ.

Инструмент Грубая кисть вносит случайные искажения («выбросы») в траекторию контура, попавшего в область действия кисти.

Инструмент Свободное преобразование позволяет масштабировать, вращать и перекашивать объект путем протягивания мыши.

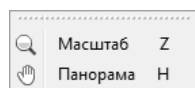


Инструмент **Обрезать** позволяет удалить области изображения за пределами выбранной.

Инструмент **Нож** позволяет разделить замкнутый контур на отдельные объекты с границами, совпадающими с траекторией разреза.

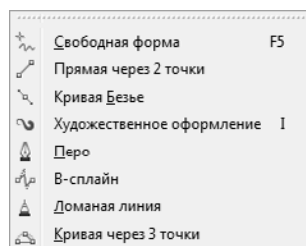
Инструментом **Ластик** удаляют часть объекта в области действия ластика, в том числе сегменты контура и внутренние области замкнутого контура.

Инструментом **Удаление виртуального сегмента** можно удалить накладывающиеся сегменты в объектах.



Инструмент **Масштаб** предназначен для выбора масштаба просмотра объектов на Рабочем поле.

Инструмент **Панорама** предназначен для перемещения изображения документа в окне Рабочего поля.



Инструмент **Свободная форма** позволяет создавать контур произвольной формы, сегменты которого образуются автоматически.

Инструмент **Прямая через 2 точки** служит для рисования прямых путем протягивания от начальной точки к конечной.

Инструмент **Кривая Безье** служит для рисования кривых Безье путем расстановки опорных точек их определения их свойств в зависимости от манипуляций с мышью.

пуляций с мышью.

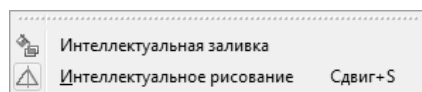
Инструмент **Художественное оформление** на самом деле содержит несколько средств: **Заготовка**, **Кисть**, **Распылитель**, **Каллиграфия**, **Нажим**. Выбор конкретного средства осуществляют (в отличие от других инструментов) на панели свойств.

Инструмент **Перо** служит для рисования кривых Безье. Отличается от рассмотренного ранее инструмента **Кривая Безье** тем, что очередной сегмент отображается на экране в интерактивном режиме, что облегчает создание сложных контуров.

Инструмент **В-сплайн** позволяет рисовать изогнутые линии, задавая их форму с помощью управляющих точек, но не разбивая кривые на сегменты.

Инструмент **Ломаная линия** обладает свойствами как пера, так и средства свободного рисования. Прямые отрезки создают щелчками мыши в опорных точках, а криволинейные сегменты — протягиванием мыши.

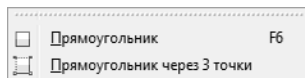
Инструмент **Кривая через 3 точки** позволяет создавать сегменты, опорные точки которых определяются протягиванием мыши между ними. Кривизна сегмента устанавливается перемещением указателя при отпущенной левой кнопке мыши. Завершается операция щелчком после достижения требуемой формы сегмента.



Инструмент **Интеллектуальная заливка** анализирует расположение объектов, соответствующее выбранной точке изображения, создает новый объект, контуры которого образованы

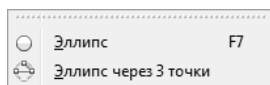
наложением объектов, и закрашивает этот объект.

Инструмент Интеллектуальное рисование преобразует мазки свободной формы в основные фигуры (прямоугольник, эллипс, треугольник и др.) или в сглаженные кривые.



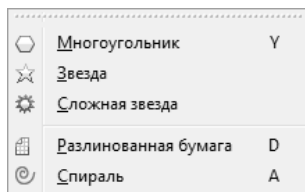
Инструмент Прямоугольник позволяет рисовать прямоугольники и квадраты методом протягивания.

Инструмент Прямоугольник через 3 точки по способу работы схож с рассмотренным ранее инструментом Кривая через 3 точки. Протягиванием мыши определяют длину и направление одной из сторон прямоугольника. Отпустив кнопку и перемещая указатель, задают длину других сторон фигуры. Щелчком мыши завершают операцию.



Инструмент Эллипс служит для рисования эллипсов и окружностей.

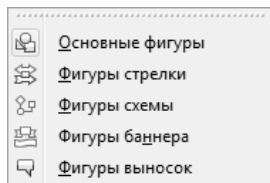
Инструмент Эллипс через 3 точки предназначен для создания эллипса методом, аналогичным рассмотренным ранее для подобных инструментов.



Инструменты Многоугольник, Звезда, Сложная звезда служат для рисования многоугольников и звезд с заданным числом углов (лучей).

Инструмент Разлинованная бумага позволяет создавать прямоугольники, разбитые на заданное число столбцов и строк.

Инструмент Спираль служит для рисования линейных и логарифмических спиралей.



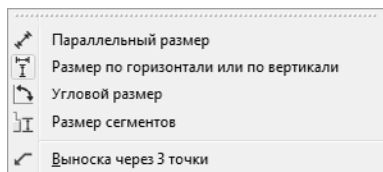
Инструменты Основные фигуры, Фигуры стрелки, Фигуры схемы, Фигуры баннера, Фигуры выносок служат для рисования соответствующих фигур из предустановленных наборов образов.



Инструмент Текст позволяет работать в двух режимах. При протягивании мыши образуется прямоугольная область, рассматриваемая как текстовый блок. При щелчке мышью текст размещается непосредственно в поле документа.



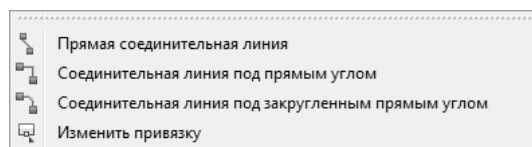
Инструмент Таблица позволяет создавать таблицы и настраивать их параметры.



Инструменты Параллельный размер, Размер по горизонтали или по вертикали, Угловой размер, Размер сегментов предназначены для полуавтоматического создания размерных линий.

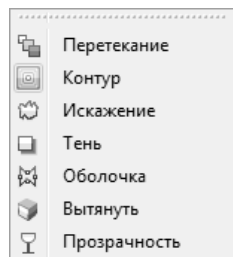
Инструмент Выноска через 3 точки предназначен для создания выносных линий и соот-

ветствующих надписей.



Инструменты Прямая соединительная линия, Соединительная линия под прямым углом, Соединительная линия под закругленным прямым углом предназначены для рисования соединительных линий в схемах.

Инструмент Изменить привязку позволяет изменить привязку соединительной линии.



Инструмент Перетекание требует наличия в документе хотя бы двух объектов. При протягивании инструмента от одного объекта к другому между ними возникают промежуточные формы, определяемые параметрами, которые заданы на панели свойств.

Инструмент Контур, воздействуя на объект, создает копии контура объекта в соответствии с параметрами, заданными на Панели свойств.

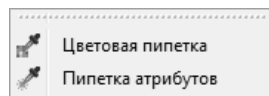
Инструмент Искажение позволяет деформировать форму объекта методом протягивания мыши.

Инструмент Тень предназначен для создания эффекта тени, отбрасываемой объектом.

Инструмент Оболочка предназначен для изменения формы объекта в соответствии с формой описанной вокруг него фигуры.

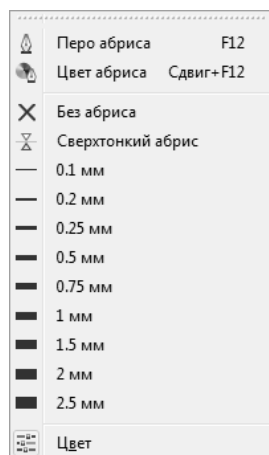
Инструмент Вытянуть позволяет создавать иллюзию глубины (объемности) объекта за счет «выдавливания» в перспективу.

Инструмент Прозрачность служит для создания различных эффектов прозрачности объекта.

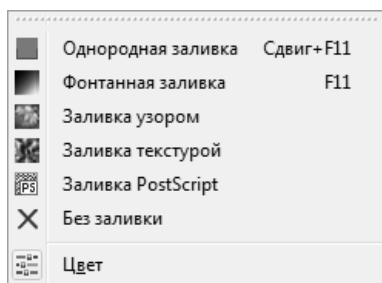


Инструмент Цветовая пипетка позволяет выбрать цвет заполнения по образцу цвета имеющегося в документе объекта.

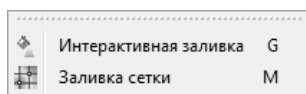
Инструмент Пипетка атрибутов позволяет скопировать различные атрибуты одного объекта для применения их к другому.



Инструменты представленного набора служат для задания свойств линий. Помимо набора стандартных значений толщин линий он имеет кнопки доступа к диалоговым окнам настройки свойств: Перо абриса и Цвет абриса. В соответствующих диалоговых окнах содержатся элементы управления всеми свойствами контура, которые доступны в программе CorelDraw.



Инструменты Однородная заливка, Фонтанная заливка, Заливка узором, Заливка текстурой, Заливка PostScript применяют для настройки различных типов заполнения объектов. Работа с этими инструментами ведется в оконном режиме, то есть для настройки используются элементы управления появляющегося при выборе инструмента диалогового окна.



Инструмент Интерактивная заливка предоставляет доступ к тем же средствам настройки свойств заполнения, что и рассмотренные выше инструменты заливки, но в интерактивном режиме — посредством манипуляций мышью и элементами управления на панели свойств.

Инструмент Заливка сетки по способу действия подобен рассмотренному выше инструменту Оболочка. Кроме изменения формы объекта, он позволяет деформировать заполнение в соответствии с параметрами сетки деформации, привязанной к объекту.

Панель свойств

После выбора любого инструмента панель свойств отображает его свойства и предоставляет набор элементов, позволяющих управлять параметрами инструмента или свойствами объекта. В связи с этим состав элементов управления на панели свойств динамически меняется в зависимости от выбранного инструмента и объекта. На рис. 16.3 вы можете видеть два варианта заполнения Панели свойств: Нет выделения), и второй, когда выбран объект Прямоугольник (Панель свойств: Прямоугольник).



Рис. 16.3. Пример динамического изменения элементов управления Панели свойств в том случае, когда ни один объект не выбран (*выше*); выбран объект Прямоугольник (*ниже*)

Если не выбран ни один объект, на панели свойств присутствуют элементы управления параметрами текущей страницы:

- ☐ раскрывающийся список Размер страницы для выбора стандартного или пользовательского формата страницы;
- ☐ счетчики Высота и ширина страницы для установки высоты и ширины страницы;
- ☐ кнопки Книжная и Альбомная для изменения ориентации страницы;
- ☐ кнопки переключения параметров текущей страницы, принятых по умолчанию для всех публикаций или только для текущего документа;

- ☐ раскрывающийся список Единицы для выбора единиц измерения;
- ☐ счетчик Расстояние перемещения для задания стандартной величины смещения узлов;
- ☐ счетчики Дублировать расстояние для установки величины смещения дубликата объекта по осям x , y .

Динамическое изменение состава элементов управления на панели свойств происходит при выборе соответствующего инструмента.

Если к объекту уже был применен какой-либо инструмент, повлиявший на его свойства, этот факт отображается в строке состояния. Например, объект, подвергшийся воздействию инструмента **Контур**, в строке состояния отображается как **Контурная группа**. Если вновь выбрать такой объект инструментом **Контур**, то можно изменить любые свойства ранее созданного контура. Такой механизм действует и применительно к другим инструментам.

Строка меню

Некоторые инструментальные средства CorelDraw доступны только из меню, если пользователь не менял заданную по умолчанию конфигурацию в окне **Параметры**. В частности, к ним относятся:

- ☐ средства обработки растровых объектов, сосредоточенные в меню **Растровые изображения**;
- ☐ средства **Настройка**, **Преобразование**, **Коррекция**, **Художественное оформление**, **Добавить перспективу**, **PowerClip**, **Ролловер** и другие, входящие в состав меню **Эффекты**;
- ☐ средства управления параметрами отображения из меню **Вид**;
- ☐ инструмент **Разъединить** из меню **Упорядочить**;
- ☐ средство записи сценариев (**Visual Basic**) из меню **Инструменты**.

Часть инструментальных средств и элементов управления можно открыть в стыкуемых палитрах. Однако принятая по умолчанию конфигурация интерфейса принципиального значения не имеет — пользователь всегда имеет возможность составить по своему усмотрению любые конфигурации панелей инструментов, меню и прочих элементов интерфейса.

Специальные средства CorelDraw

О программе CorelDraw традиционно сложилось представление как о мощном и удобном средстве создания рекламных материалов, логотипов, иллюстративной и художественной графики. Действительно, именно в этих областях программа имеет самое широкое распространение. Однако не следует забывать о некоторых возможностях программы, превращающих ее в эффективное средство разработки инженерной и научной документации. С помощью CorelDraw можно создавать точные машиностроительные и архитектурные чертежи, строить графики, иллюстрировать учебные материалы. Для целей точного позиционирования объектов относительно рабочего листа и друг друга, управления проектом в целом и его отдельными элементами служат специальные средства CorelDraw.

Диалоговое окно **Выровнять и распределить** открывается по команде **Упорядочить** ► **Выровнять и распределить** ► **Выровнять и распределить** (рис. 16.4).

На вкладке **Выравнивание** предлагаются средства выравнивания объектов относительно центра или края страницы, а также относительно других объектов или опорных точек направляющей сетки. На вкладке **Распределение** задают равномерное распределение объектов относительно других объектов или страницы документа. Управление выравниванием и распределением, в сочетании с возможностью явно задавать размеры и положение объектов на Панели свойств, позволяет выполнять чертежи с точностью до третьего знака после запятой, что вполне достаточно для подавляющего большинства инженерных задач.

Очень удобно использовать в работе стыкуемые палитры (окна настройки). Их полный список приведен в меню **Окно** ► **Окна настройки**. Пользователь сам выбирает те палитры, которые следует отобразить в текущий момент. Открывают окно палитры щелчком на ее закладке, а закрывают щелчком на кнопке **Свернуть окно настройки**. Удаляют палитру щелчком на кнопке **Закрыть**. Любую палитру можно вывести в отдельное окно, перетаскив за корешок в область рабочего поля.

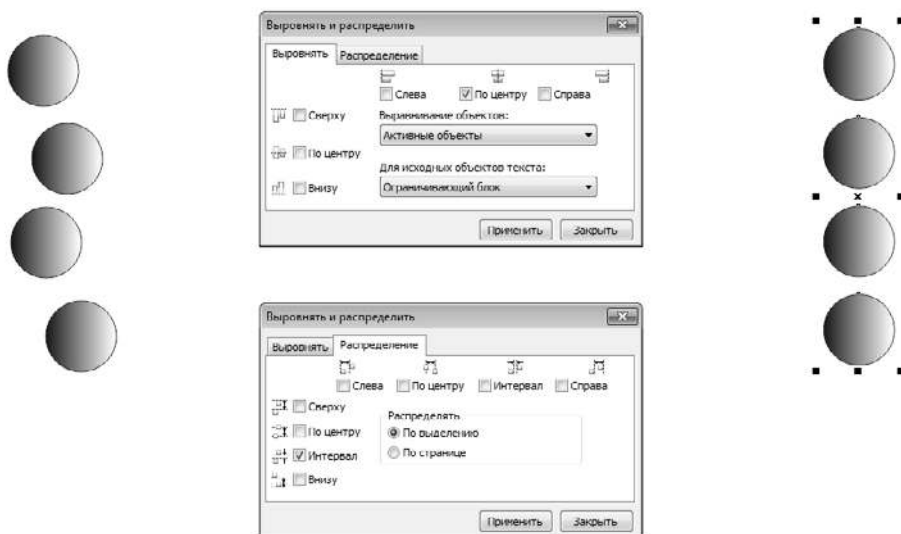
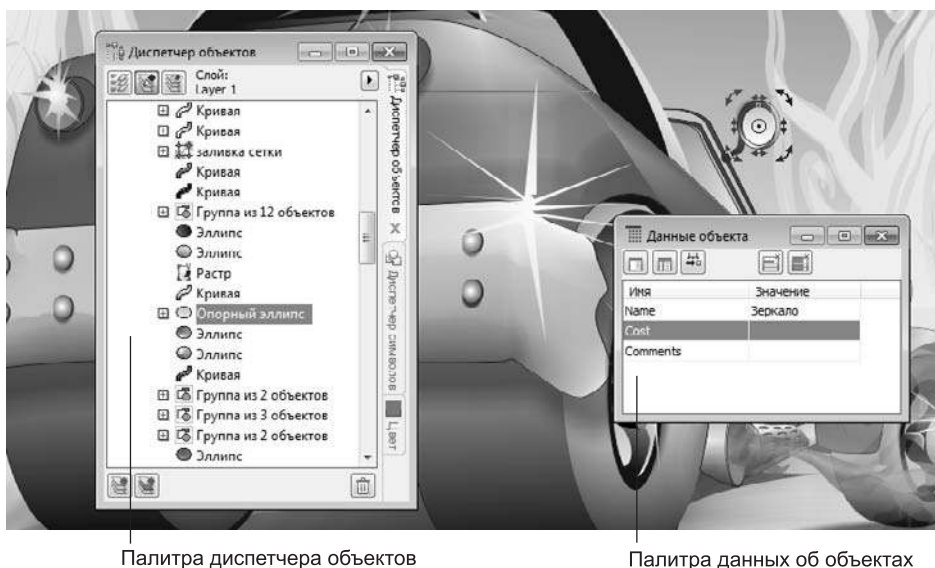


Рис. 16.4. Пример использования средства выравнивания и распределения для конфигурирования группы объектов

При разработке сложных проектов, когда число объектов в документе превышает несколько тысяч, особую важность приобретает использование двух специфических палитр: **Диспетчер объектов** и **Данные объекта** (рис. 16.5).

Стыкуемая палитра **Диспетчер объектов** является мощным средством управления положением объектов и контроля над их свойствами. Ее открывают из меню **Инструменты** ► **Диспетчер объектов**. В этой палитре представлено послойное расположение всех объектов с указанием их названий и свойств. Переход к любому объекту означает его автоматическую выборку с динамическим изменением элементов управления на панели свойств.



Палитра диспетчера объектов

Палитра данных об объектах

Рис. 16.5. Пример использования менеджера объектов и менеджера сведений об объектах при создании сложного проекта

Дополнительные возможности по управлению сложными проектами предоставляет стыкуемая палитра **Сведения об объектах**, где любому объекту можно присвоить собственное имя, назначить поля для его характеристик (например, используемых материалов, стоимости работы и пр.). В целом средства управления проектом и объектами позволяют создать сложные многоуровневые документы и организовать совместную работу по их разработке.

16.3. Рисование графики

Создание и модификация объектов

Приемы рисования типовых объектов в CorelDraw мало отличаются от таковых в других векторных редакторах. Поэтому мы рассмотрим в основном возможности по манипуляции свойствами объектов и приемы их модификации стандартными средствами программы.

Все объекты, создаваемые инструментами программы, обладают рядом атрибутов и общими возможностями модификации. При выборе любого объекта инструментом **Выбор** на панели свойств отображаются:

- ☐ поля с параметрами его координат (x, y);
- ☐ поля с параметрами размеров по горизонтали и вертикали;
- ☐ поля для установки масштаба (в процентах);
- ☐ кнопка разрешения (запрещения) пропорционального масштабирования;
- ☐ поле для задания угла поворота;
- ☐ кнопки инструментов зеркального отражения по вертикали и горизонтали.

Состав прочих элементов управления зависит от типа конкретного объекта.

Операции с графикой

Свойства линии. Для линий, создаваемых с помощью инструментов Свободная форма, Перо, Ломаная линия, Кривая через 3 точки и других, до начала рисования на панели свойств могут быть заданы: форма наконечников, тип линии, ее толщина, степень сглаживания.

Контур. Все линии представляют собой контур (открытый либо замкнутый). Инструментами Кривая Безье или Перо рисуют кривую линию (протягиванием мыши в начальной и конечной точках сегмента) или прямую (щелчками в начальной и конечной точках). При нажатой клавише CTRL для прямой можно задать фиксированный (с шагом 15°) угол наклона. Выбрав полученный объект инструментом Выбор по окончании рисования, получают доступ к редактированию его свойств.

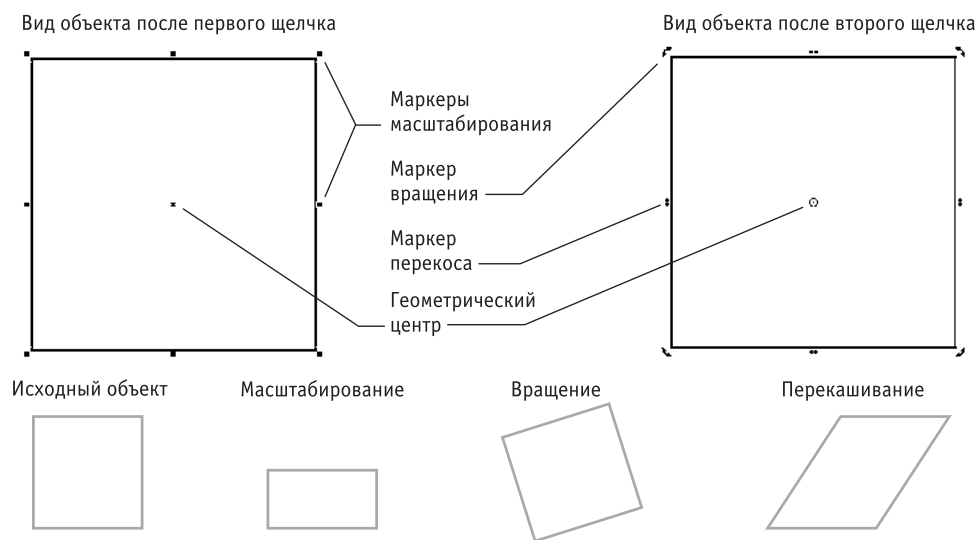


Рис. 16.6. Пример операций с объектом с использованием инструмента Выбор

Для открытой линии специфическим инструментом является средство Замкнуть кривую (доступно на Панели свойств кривой), соединяющее конечные точки и превращающее линию в замкнутый контур.

Узлы. Если объект выбран инструментом Фигуры, открываются возможности модификации его узловых точек. Для этого выбирают одну из точек и либо перемещают ее в нужное положение, либо воздействуют на управляющие касательные. Допустимые операции отображены на панели свойств. К ним относятся:

- ☐ добавление или удаление узловых точек: Добавить узлы или Удалить узлы;
- ☐ объединение двух точек: Соединить два узла;
- ☐ разрыв линии в выбранной точке: Разъединить кривую;
- ☐ преобразование кривой в прямую, и наоборот: Преобразовать в прямую и Преобразовать в кривую;
- ☐ изменение формы кривой в данной точке: Перегиб, Сглаженный узел, Симметричный узел;

- ❑ изменение направления линии: Обратить направление;
- ❑ преобразование кривой в замкнутый контур путем соединения выбранных последовательно (с использованием клавиши SHIFT) конечных точек: Замкнуть кривую;
- ❑ вычленение отдельной кривой из объекта, представляющего собой комбинацию сегментов: Извлечь фрагмент;
- ❑ автоматическое преобразование кривой в замкнутый контур: Замкнуть кривую;
- ❑ преобразование кривой путем масштабирования, вращения и перекашивания сегмента, выравнивания узлов: Растянуть или масштабировать узлы, Rotate Повернуть или наклонить узел, Выровнять узлы;
- ❑ включение режима «эластичных» (плавных) преобразований: Гибкий режим.
- ❑ выбор всех узлов: Выбрать все узлы.
- ❑ показ или скрытие ограничивающего блока при использовании кривых: Ограничивающий блок.

Кривые Безье. Для линии, создаваемой с помощью инструмента Кривая Безье, до начала рисования могут быть заданы те же параметры, что и в предыдущем случае, за исключением параметра сглаживания. Приемы рисования: прямые линии создают щелчками в опорных точках (при нажатой клавише CTRL — с фиксированным углом наклона), для рисования кривых необходимо после щелчка слегка протянуть мышью от опорной точки (рис. 16.7). Дальнейшие способы модификации полученного объекта аналогичны изложенным ранее.

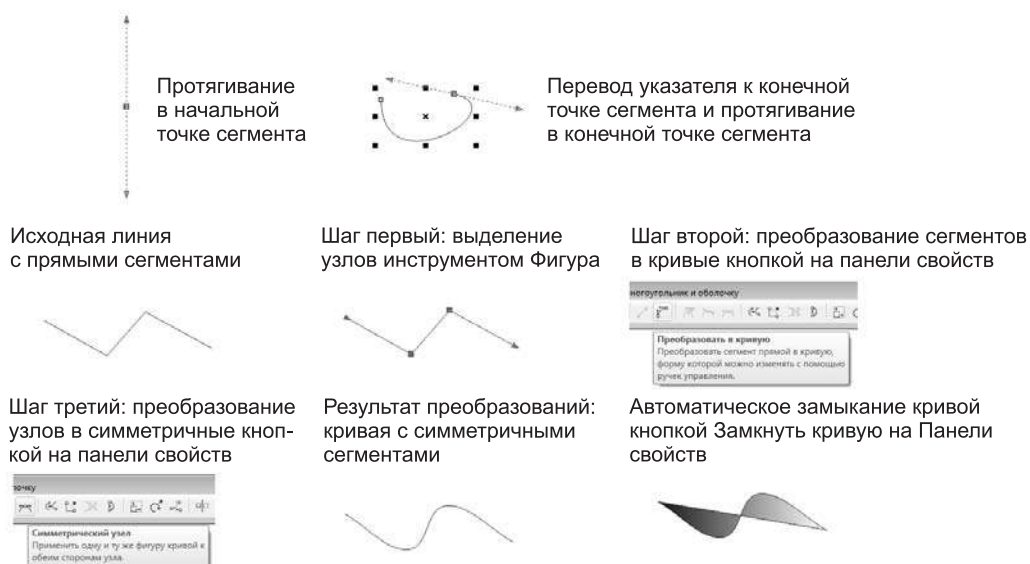


Рис. 16.7. Приемы рисования и редактирования контуров различными средствами

Художественные средства рисования. Инструмент Художественное оформление имеет ряд интересных особенностей, позволяющих создавать своеобразные

объекты. После щелчка на кнопке инструмента на панели свойств отображаются установки, принятые по умолчанию. То есть до начала рисования инструментом считается перо (имитирующее плакатное), для которого можно установить степень сглаживания рисуемой линии, ее толщину и тип. Справа от кнопки заготовленных типов перьев последовательно расположены кнопки альтернативных инструментов: кисть, распылитель, каллиграфическое перо, перо с нажимом. Последний инструмент работает только в сочетании с графическими планшетами, поддерживающими такую функцию.

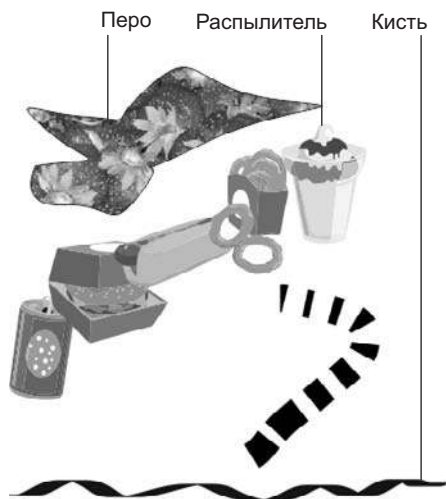


Рис. 16.8. Примеры объектов, созданных средствами Artistic Media

При выборе инструмента Кисть появляется возможность выбора типа кисти, при выборе каллиграфического пера — выбора угла его наклона.

При выборе инструмента Распылитель на панели свойств появляются дополнительные элементы, в том числе:

- ☐ поля установки размеров элементов заполнения (по вертикали и горизонтали, в процентах от исходного размера) с кнопкой разрешения (запрещения) пропорционального масштабирования;
- ☐ список выбора типа заполнения;
- ☐ кнопки сохранения или удаления типа заполнения;
- ☐ список выбора порядка заполнения;
- ☐ кнопка добавления модифицированного заполнения к используемому типу;
- ☐ кнопка для вызова диалогового окна, позволяющего модифицировать тип заполнения;
- ☐ счетчики установки интервала между столбцами и рядами элементов заполнения;
- ☐ кнопка для изменения угла поворота элементов заполнения;
- ☐ кнопка для выбора параметра смещения элементов заполнения;

- ❑ кнопка масштабирования;
- ❑ кнопка сброса всех внесенных изменений и возврата к значениям по умолчанию для данного типа заполнения;
- ❑ кнопка включения и отключения отображения ограничивающего блока.

Свойства элементов оформления. Элементы заполнения в инструменте Кисть являются едиными объектами, и модифицировать можно только объект в целом. Объекты, созданные инструментом Распылитель, можно разбивать на отдельные элементы: Упорядочить ► Разъединить, Упорядочить ► Отменить группировку — и затем работать с каждым элементом как с отдельным объектом.

Размерные линии. Перед простановкой размерных линий нужно выбрать инструмент, соответствующий типу этих линий:

Параллельный размер, Размер по горизонтали или по вертикали, Угловой размер, Размер сегментов, Выноска через 3 точки.

В Панели свойств каждого из инструментов можно указать формат представления данных измерений, точность представления данных, единицы измерения. В полях Префикс и Суффикс при необходимости проставляют текстовые или цифровые данные. С помощью соответствующих элементов управления настраивают параметры линий. Кнопкой Динамическая простановка размеров включают (отключают) режим динамического представления размещения данных во время выбора опорных точек. Выделив текст, созданный автоматически при создании размерных и выносных линий, можно настроить его параметры средствами динамически изменяющейся Панели свойств.

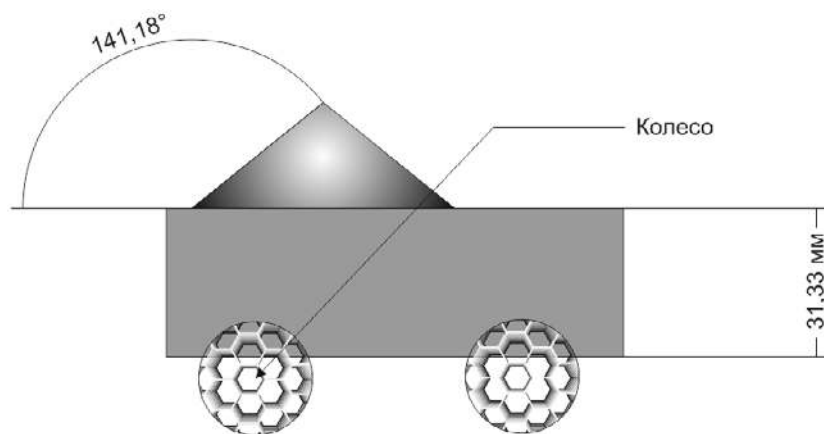


Рис. 16.9. Простановка размерных линий

Прямоугольники. С помощью инструмента Прямоугольник создают прямоугольники. До начала рисования можно задать радиусы скругления любого из углов прямоугольника счетчиками Закругленный угол. Там же есть возможность настроить параметры углов, используя средства Угол с выемкой и Угол с фаской. Прямоугольник является объектом, представляющим собой замкнутый контур. Поэтому

на панели свойств появляются новые элементы, характерные для всех замкнутых контуров. Первым из них является раскрывающаяся кнопка **Стиль обтекания**, открывающая доступ к кнопкам для выбора варианта размещения текстового блока внутри замкнутого контура. Вторым элементом является кнопка **Преобразовать в кривые**, преобразующая прямоугольник в обычный набор сегментов кривой. После такого преобразования узловые точки можно модифицировать так же, как в любом другом контуре.

Эллипсы. Инструментом **Эллипс** рисуют эллипсы (и окружности как их частный случай). Приемы рисования аналогичны приемам рисования прямоугольника. По завершении рисования на панели свойств появляются новые элементы управления свойствами, характерные для эллипсов. Кнопка **Сектор** позволяет интерактивно или изменением значений счетчиков **Начальный** и **Конечный угол** задавать параметры сектора, вырезаемого из эллипса. Кнопкой **Дуга** таким же образом задают параметры вырезаемой дуги. Кнопкой **Изменить направление** управляют размещением сектора (дуги) относительно узловых точек.

Прочие фигуры. Другие средства рисования позволяют создавать сложные и составные объекты: многоугольники, звезды, разграфленные прямоугольники, спирали, заранее подготовленные фигуры. Элементы управления свойствами таких объектов размещены на панели свойств и достаточно просты для понимания. Методы создания не отличаются от общепринятых: объект рисуют протягиванием мыши, а для образования правильной фигуры удерживают клавишу **CTRL**.

Заранее подготовленные фигуры отличаются от остальных объектов тем, что имеют в составе дополнительное средство управления — так называемый **глиф** (*glyph*). С его помощью можно изменять форму контура либо объекта в целом, либо его части.

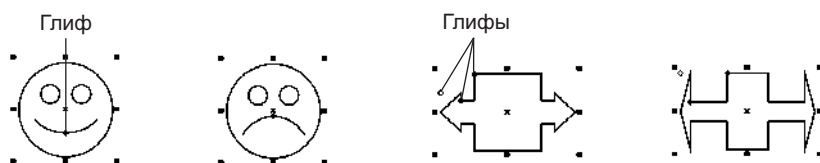


Рис. 16.10. Использование глифов для изменения формы фигур

Копирование, дублирование и клонирование

При создании однотипных объектов удобно использовать операции копирования, дублирования и клонирования. Операция копирования относится к числу стандартных для всех приложений Windows: выбранный объект копируется в буфер обмена командой **Правка ▶ Копировать** и вставляется в документ командой **Правка ▶ Вставить**. После вставки никакой связи свойств нового объекта с исходным не сохраняется, и он рассматривается как полностью независимый объект.

Операция дублирования (**Правка ▶ Дублирование**) создает объект, идентичный исходному. Величину смещения дубликата относительно исходного объекта задают счетчиками **Смещение** на панели свойств (счетчик появляется, если нет выбран-

ных объектов). При дублировании выборка (фокус) автоматически смещается на дубликат. Если в это время изменить некоторые свойства дубликата (например, координаты, угол поворота, размер), не снимая выборки, то при следующей операции дублирования соответствующие параметры пропорционально изменятся у следующего дубликата. Таким методом удобно создавать группы объектов с фиксированным шагом масштабирования, углом поворота и заданным смещением относительно друг друга.

Свойства контура и заполнения дубликата теряют связь с соответствующими параметрами исходного объекта и поэтому могут быть изменены независимо друг от друга у любого экземпляра.

Операция клонирования (Правка ► Клонирование) отличается от дублирования двумя особенностями. Во-первых, клон наследует свойства контура и заполнения исходного объекта и сохраняет связь с этими свойствами: изменение свойств исходного объекта вызывает соответствующие изменения у клона. Во-вторых, операция клонирования «одноступенчатая»: нельзя создать клон от клона.

16.4. Заполнение объектов

Неотъемлемым свойством замкнутых контуров (а в некоторых случаях и открытых) является метод заполнения (заливки). Простейшим способом выбора заполнения служит щелчок левой кнопкой на выбранном элементе цветовой палитры. Щелчком правой кнопкой мыши выбирают цвет контура. Следующий прием заключается в выборе объекта, открытии щелчком правой кнопкой мыши контекстного меню, выборе пункта Свойства и выборе в открывшемся диалоговом окне вкладки Заливка, где можно указать метод заливки. Наконец, можно открыть одно из диалоговых окон управления параметрами заливки щелчком на кнопке инструмента настройки заливки на Панели инструментов и настроить параметры заливки.

Средства заполнения. Заливку замкнутого контура удобнее всего выполнять инструментом Интерактивная заливка. Например, для градиентного заполнения достаточно выбрать нужный цвет, установить тип заливки в Панели свойств инструмента и протянуть курсор внутри контура в направлении действия градиента.

Сплошное заполнение. Для заполнения сплошным цветом можно выбрать цветовую модель в раскрывающемся списке Тип сплошной заливки, а затем счетчиками выставить отдельные параметры, соответствующие типу цветовой модели. Например, для модели *СМУК* это процентные соотношения цветов ее палитры.

Градиентное заполнение. Для выполнения градиентной (в переводе Corel-DRAW X5 эта заливка называется «фонтанной») следует выбрать один из предустановленных типов заливки:

- ☐ Линейные;
- ☐ Радиальные;
- ☐ Конические;
- ☐ Квадрат.

После выбора типа заливки на Панели свойств становятся доступными элементы управления для настройки параметров градиента. В частности, это цвета градиента, средняя точка, угол и сдвиг края.

После создания градиентной заливки ее параметры можно менять интерактивно путем перетаскивания элементов управления градиентом прямо на объекте.

Заполнение узором. В случае выбора метода заполнения узорами, в частности это Двухцветный узор, Полноцветный узор, Растровый узор, появляются следующие элементы управления:

- ☐ раскрывающийся список с образцами моделей заполнения;
- ☐ кнопки выбора цветов переднего и заднего плана Цвет переднего плана и Цвет заднего плана, которые активны только для модели заливки Двухцветный узор;
- ☐ кнопки для выбора размеров элементов заполнения: Малая плитка, Средняя плитка, Крупная плитка;
- ☐ счетчик изменения размеров элементов заполнения по вертикали и горизонтали Изменить плитку;
- ☐ кнопка для включения (отключения) режима трансформации заполнения при модификации объекта Преобразовать вместе с объектом;
- ☐ кнопка Зеркальная заливка плиткой, которая позволяет упорядочить элементы заливки таким образом, чтобы чередующиеся элементы были отражением друг друга;
- ☐ кнопка выбора объектов для их последующего использования в качестве узоров Создать узор;
- ☐ кнопка Скопировать свойства, которая позволяет применить свойства выбранной заливки одного объекта к другому объекту.

После создания заполнения узором его можно редактировать в интерактивном режиме путем перетаскивания элементов управления на объекте.

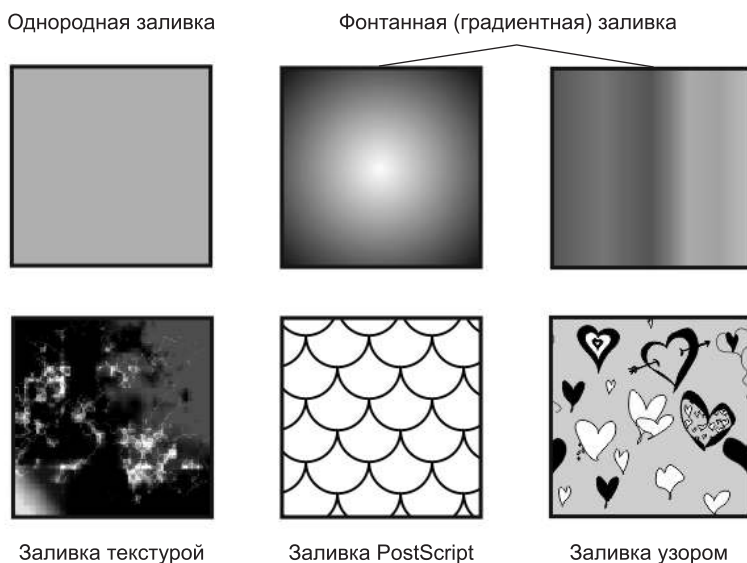


Рис. 16.11. Примеры объектов с заполнением различных типов

Заполнение текстурами. Метод заполнения текстурами *Заливка текстурой* предлагает элементы управления в виде раскрывающегося списка библиотеки образцов Библиотека текстур, раскрывающегося списка выбора образцов Первый цвет заливки, кнопки Преобразовать вместе с объектом, кнопок Зеркальная заливка плиткой, Пересоздать текстуру, Скопировать свойства и Параметры текстуры. Последняя кнопка открывает доступ к одноименному диалоговому окну, где устанавливаются разрешение (в точках на дюйм) текстуры и ограничения по размерам элемента текстуры (в пикселях) и объема ее файла. После создания заполнения его редактируют способами, описанными выше.

Заполнение PostScript. При выборе метода заполнения *Заливка PostScript* появляется единственный элемент управления — раскрывающийся список для выбора образцов текстур *Текстуры заливки PostScript*. Интерактивное редактирование текстур PostScript недоступно. При необходимости изменить параметры такой текстуры можно воспользоваться инструментом *Заливка PostScript*, открывающим диалоговое окно *Текстура PostScript* и счетчиками изменить параметры заливки.

Редактирование заполнения. Инструмент *Заливка сетки* предназначен для интерактивного редактирования заполнения, уже присвоенного объекту. При выборе инструмента на панели свойств появляются элементы управления в виде счетчиков *Размер сетки*, с помощью которых текстура разбивается на необходимое число участков. Точки, располагающиеся на пересечении границ участков, обладают всеми свойствами стандартных узловых точек, и операции над ними идентичны таковым для опорных точек. Тем самым можно гибко менять структуру заполнения, добиваясь необходимого эффекта.

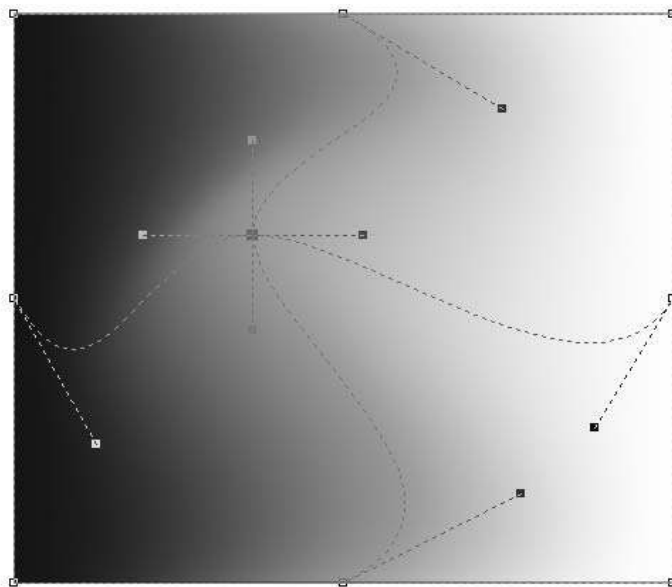


Рис. 16.12. Использование сетки деформации

Редактирование прозрачности. Инструмент Прозрачность предназначен для динамического управления степенью прозрачности объекта. При его выборе на панели свойств появляется кнопка Изменение прозрачности, раскрывающийся список для выбора метода заполнения Тип прозрачности и список Действие прозрачности, который позволяет настроить взаимодействие прозрачного цвета с элементами, над которыми он находится.

С помощью движка Средняя точка прозрачности управляют степенью прозрачности. Кнопкой Заморозить закрепляют сделанные изменения, а кнопкой Удалить прозрачность отменяют все модификации.

16.5. Операции с текстом

Особое место среди средств CorelDraw занимает инструмент Текст. В то время как все другие инструменты работают с графическими объектами, текст не является графикой и потому обладает специфическими свойствами (например, начертанием шрифта и пр.). Однако при необходимости любой текстовый объект может быть преобразован в графический: Упорядочить ► Преобразовать в кривую. А вот обратное преобразование невозможно.

Типы текстовых объектов. В CorelDraw текстовые объекты бывают двух типов. Первый тип текстовых объектов называется Фигурный текст и представляет собой свободно размещаемые строки текста. Второй тип называется Простой текст и отличается размещением символов внутри текстового блока. Форму и границы блока можно модифицировать стандартными для графических объектов средствами, однако символы всегда будут находиться внутри него, а их редактирование (изменение свойств) возможно только инструментом Текст. Отличие объектов типа Фигурный текст состоит в том, что их можно модифицировать как графические объекты (например, вращать или растягивать) с воздействием преобразований на сами символы, что недопустимо для объектов блочного текста, модификация которого производится для блока в целом.

Средство конвертирования текстовых объектов Текст ► Преобразовать... позволяет осуществлять взаимные преобразования между двумя типами текстовых объектов. При этом все изменения, внесенные в объект Фигурный текст средствами редактирования графики и повлиявшие на форму символов (за исключением размера шрифта), автоматически отменяются во время преобразования в объект Простой текст.

Набор и форматирование. Средства набора, форматирования и редактирования текста находятся в палитре свойств и в раскрывающемся меню Текст строки меню. Их состав аналогичен стандартным средствам текстовых редакторов и программ верстки. Однако имеются два специфических инструмента, не встречающиеся в специализированных программах обработки текста.

Инструмент Текст вдоль пути позволяет расположить текст вдоль линии. Для этого достаточно последовательно выбрать объект типа Фигурный текст и любой графический объект и применить инструмент. Можно также подвести текстовый курсор к кривой так, чтобы рядом с вертикальной чертой курсора появилась волнистая линия, после чего щелкнуть мышью и начать ввод текста. После создания

совмещенного текстово-графического объекта на панели свойств появляются инструменты редактирования:

- ☐ раскрывающийся список Ориентация текста;
- ☐ счетчики Расстояние от пути, Смещение;
- ☐ кнопки Отразить текст по горизонтали, Отразить текст по вертикали;
- ☐ другие элементы управления, предназначенные для форматирования текста.

Редактировать текстовое содержимое можно средствами инструмента Текст.

Другим специфическим инструментом работы с текстом является средство Текст ► Простой текст ► Текст в рамку, позволяющее поместить текстовый блок внутри замкнутого контура. При этом границы контура рассматриваются как границы блока. Для размещения текста внутри замкнутого контура достаточно выбрать инструмент Текст, подвести текстовый курсор к границе контура так, чтобы рядом с вертикальной чертой курсора появился значок в виде символов АВ в рамке. После щелчка мышью можно начинать ввод текста.

16.6. Изменение формы объектов

Эффективными средствами изменения формы объектов выступают инструменты, объединенные в группе интерактивных инструментов (Перетекание, Контур), и некоторые другие.

Смешивание (перетекание). Для использования инструмента Перетекание необходимо наличие в документе как минимум двух объектов (не считая текстовых). После выбора инструмента (но до его применения) на панели свойств устанавливаются:

- ☐ число промежуточных трансформаций с помощью счетчика Перетекание объектов;
- ☐ направление трансформации кнопками Прямое перетекание, Перетекание по часовой стрелке, Перетекание против часовой стрелки;
- ☐ параметры трансформации кнопками Ускорение объекта и цвета, Ускорение размера.

Применение инструмента заключается в выборе первичного объекта и протягивании мыши к вторичному объекту. Промежуточные объекты трансформации создаются автоматически. После завершения операции на панели свойств становятся доступны следующие элементы управления:

- ☐ счетчик Направление перетекания;
- ☐ кнопка Перетекание с поворотом;
- ☐ раскрывающиеся кнопки Дополнительные параметры перетекания, Начальный и конечный объект, Свойства пути, с помощью которых редактируют свойства полученного объекта.

Контур. Инструмент Контур служит для создания копий контура объекта. Копии могут расходиться из объекта или входить в объект.

Свободная деформация. Инструмент Искажение служит для свободной деформации объектов. После выбора инструмента на панели свойств появляются кнопки

выбора типа деформации: Потянуть углы, Сжать углы, Кручение, Штамп, Застежка-молния. В зависимости от выбранного типа деформации могут присутствовать и другие элементы управления.

Например, для средства Потянуть углы это счетчик установки размаха деформации Амплитуда сжатия и растяжения. После применения инструмента становится доступной кнопка включения деформации относительно центра объекта Искажение по центру.

Оболочка. Инструмент Оболочка позволяет модифицировать выбранные объекты в соответствии с формой огибающего их контура. То есть вокруг объекта размещается управляющий объект, имеющий собственные опорные точки, перемещение которых вызывает перемещение расположенных вблизи узловых точек модифицируемого объекта. После выбора инструмента на панели свойств появляются элементы управления:

- ☐ список выбора готовых форм огибающих: Круговая, Прямолинейный, Прямолинейный наклон, Вдавливание, Нажатие, Выдавливание;
- ☐ кнопки Добавить заготовку и Удалить заготовку для управления списком готовых форм огибающих;
- ☐ кнопки Режим прямой, Режим одной дуги, Режим двойной дуги и Произвольный режим для выбора режима огибающей;
- ☐ кнопки управления узлами огибающей;
- ☐ кнопка Сохранить прямые и другие элементы управления.

Модификация объекта в соответствии с формой огибающей заключается в перетаскивании мышью выбранной узловой точки и настройке параметров на панели свойств.

Вытягивание. С помощью инструмента Вытягивание создают псевдотрехмерные объекты путем «вытягивания» контуров (открытых или замкнутых). После выбора инструмента на панели свойств доступен список заготовок. Инструментом можно начинать работать немедленно после выбора, не выбирая заготовку. После того как эффект применен к объекту, появляется возможность дальнейшей работы. Так, на панели свойств становятся доступными следующие инструменты:

- ☐ счетчики Позиция объекта;
- ☐ список Тип вытягивания;
- ☐ счетчик Глубина;
- ☐ счетчики Координата точки схода;
- ☐ список Свойства точки схода;
- ☐ кнопки Поворот вытягивания, Цвет вытягивания, кнопки управления освещением вытягивания и некоторые другие.

Тень. С помощью инструмента Тень создают эффект тени, отбрасываемой объектом. После выбора инструмента необходимо протянуть курсор в направлении, куда будет помещена тень. По завершении операции на панели свойств становятся доступными элементы управления свойствами тени.

Нож. Инструмент Нож служит для расчленения контуров на составляющие. Если пересечь контур инструментом, в точке пересечения образуется разрыв, то

есть формируются две узловые точки, принадлежащие разным сегментам. Инструмент применим как к открытым, так и к замкнутым контурам. При выборе инструмента на палитре свойств появляется кнопка **Сохранить как один объект**, управляющая режимом представления модифицируемого объекта как единого целого или с образованием независимых объектов, и кнопка **Закрывать** автоматически при вырезании, управляющая режимом автоматического соединения разомкнутых контуров.

Ластик. Инструмент Ластик применим как к контурам, так и к заливкам объектов. В первом случае контур, если он не пересекается инструментом, как бы выдавливается по границе поля инструмента, а при пересечении разрывается. Во втором случае свойства заливки в области действия поля инструмента изменяются. Вокруг очищенной области создается собственный замкнутый контур. Таким образом, инструмент образует отдельный объект, вложенный внутрь модифицируемого. При выборе инструмента на палитре свойств появляется счетчик установки поля действия **Толщина ластика**, кнопка выбора режима действия **Уменьшить число узлов**, кнопка переключения формы поля инструмента между круглой и квадратной.

Удаление виртуального сегмента. Этот инструмент позволяет удалить накладывающиеся сегменты в объектах.

Обрезка. Инструмент Обрезка позволяет обрезать части объектов, находящиеся за рамкой выделения, нарисованной с помощью этого инструмента. После того как рамка нарисована, есть возможность настроить ее координаты, размеры и поворот, а также удалить рамку, воспользовавшись кнопкой с соответствующим названием. Для применения эффекта обрезки нужно нажать на клавишу **Enter** или сделать двойной щелчок в рамке обрезки.

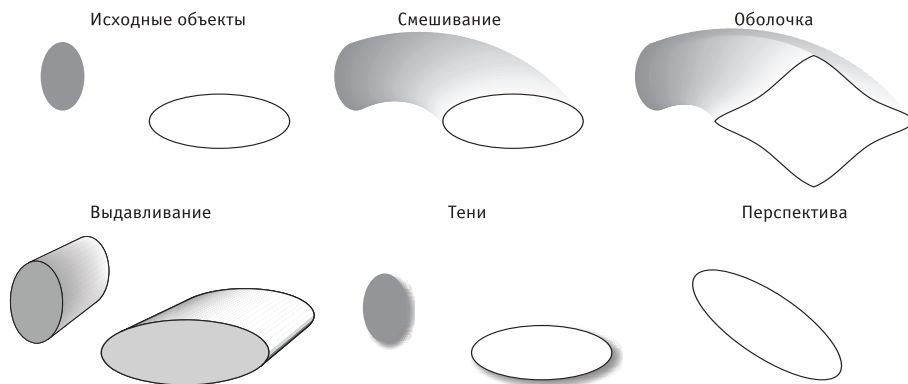


Рис. 16.13. Примеры использования некоторых эффектов

Линза. Средство модификации Линза доступно из меню **Эффекты** ▶ **Линза**. По указанной команде открывается окно настройки (палитра) инструмента. В палитре объединены элементы управления для различных эффектов, применяемых к выбранному объекту. Например, к ним относится эффект «рыбьего глаза», то есть увеличения центральной части объекта с уменьшением коэффициента масштабирования по краям.

Перспектива. Средство создания эффекта перспективы доступно из меню: Эффекты ► Добавить перспективу. После его выбора на объект накладывается прямоугольная сетка из вспомогательных линий, имеющая в углах узловые точки. Инструментом Фигура перемещают узловые точки, добиваясь необходимого эффекта перспективы. При нажатой клавише CTRL перемещение происходит строго вдоль одной из осей вспомогательной сетки. При нажатом сочетании клавиш CTRL+SHIFT происходит одновременное одинаковое смещение узлов, расположенных на одной оси.

16.7. Операции с группами

К операциям над группами объектов относятся группировка Упорядочить ► Сгруппировать, Упорядочить ► Объединить, а также логические операции пересечения (логическое «И»), отсечения (логическое вычитание), объединения (логическое «ИЛИ»). Логические операции доступны из строки меню: Упорядочить ► Формирование или из стыкуемой палитры Формирование.

Группировка. Группировка объектов заключается в их выборке инструментом Выбор с последующей фиксацией взаимного расположения командой Сгруппировать. При этом свойства сгруппированных объектов не меняются. Допустима группировка с многократным вложением групп друг в друга. Обратная операция происходит пошагово командой Отменить группировку или сразу над всеми отдельными объектами, сколько бы уровней вложения групп не существовало — командой Отменить группировку полностью. Все команды группировки (разгруппировки) доступны на панели свойств или из меню Упорядочить.

Объединение. Объединение — Упорядочить ► Объединить — отличается от группировки тем, что в результате операции создается единый итоговый объект, наследующий свойства объекта, выбранного последним. Даже если объекты не соприкасаются друг с другом, комбинирование приводит к присвоению итоговому объекту свойств последнего объекта. Если объекты пересекаются, в зоне пересечения свойства заполнения меняются на значение «без заполнения». Обратная операция — Упорядочить ► Разъединить — приводит к разъединению объектов с сохранением присвоенных им во время операции комбинирования свойств. Однако зона пересечения при этом вновь обретает исходные свойства заполнения, так как принадлежит уже разным объектам. Команда комбинирования доступна на панели свойств, команда разъединения — только из меню Упорядочить.

Логические операции. Управление логическими операциями над объектами удобнее осуществлять из стыкуемой палитры Формирование. Однако эти средства доступны также из меню: Упорядочить ► Формирование.

Операция ИЛИ. Операция Объединение соответствует логической операции «ИЛИ». Выбранные объекты образуют единый контур, свойства которого наследуются от объекта, выбранного последним. Если выбранные объекты не пересекаются, образуется итоговый единый объект, не имеющий общего контура, но также наследующий свойства последнего выбранного объекта. Если объекты не пересекались, возможна обратная операция Упорядочить ► Разъединить, но исходные свойства объектов при этом не восстанавливаются.

В палитре Формирование необходимо из списка вариантов формирования объектов выбрать Объединение. В окне будет показан образец результата операции. Флажками Исходные объекты и Конечные объекты в разделе Сохранять оригинал можно выставить режим сохранения копий исходных объектов.

Операция вычитания. Операция Исклучение соответствует логическому вычитанию. Выбранные объекты должны иметь зону пересечения, потому что именно она вычитается из итогового объекта, выбранного последним. Исходные объекты не объединяются, их свойства не передаются. Объект-источник (выбранный первым) вообще не претерпевает никаких изменений. Обратная операция невозможна, и восстановление исходного состояния доступно только путем отмены команды: Правка ► Отменить.

В палитре Формирование необходимо выбрать операцию Исклучение. В окне будет показан графический пример итога операции. Флажками Исходные объекты и Конечные объекты в разделе Сохранять оригинал можно выставить режим сохранения копий исходных объектов.

Частными случаями операции вычитания служат команды Передние минус задние, Задние минус передние.

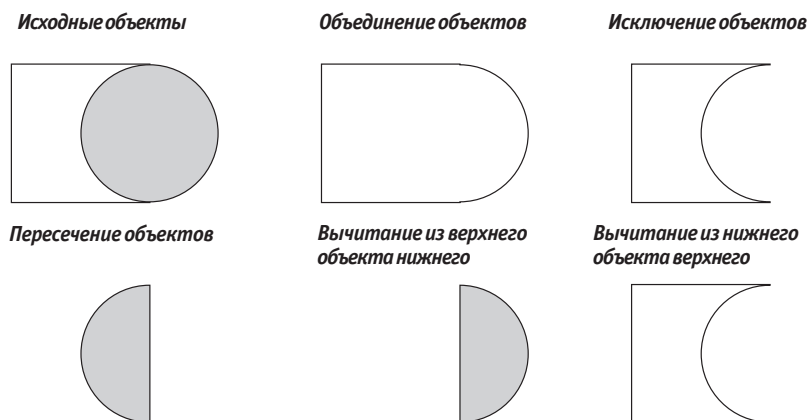


Рис. 16.14. Примеры логических операций

Операция И. Операция Пересечение соответствует логической операции «И». Выбранные объекты должны иметь зону пересечения, потому что именно она остается в качестве итогового объекта. Исходные объекты исчезают. Итоговый объект наследует свойства объекта, выбранного последним. Обратная операция невозможна, и восстановление исходного состояния доступно только путем отмены команды: Правка ► Отменить.

В палитре Формирование необходимо выбрать операцию Пересечение. В окне будет показан графический пример итога операции. Флажками Исходные объекты и Конечные объекты в разделе Сохранять оригинал можно выставить режим сохранения копий исходных объектов.

Объекты PowerClip. В программе CorelDraw существует особый класс объектов, называемый PowerClip (Обрезка). Такие объекты представляют собой объект-контейнер, в который могут быть вложены другие объекты в качестве содержимого.

Допустимо многоуровневое вложение, то есть в любой объект-контейнер можно вложить другие контейнеры со своим содержимым. Все вложенные объекты сохраняют свои свойства, их можно редактировать любыми инструментами, извлекать из контейнера или перемещать в другой контейнер. Если контейнер превышает по размерам вложенный объект, последний отображается на фоне контейнера. В противном случае отображается только та часть вложения, которая помещается в рамки контейнера. Поверх отображается объект, размещенный последним. Распознать объект PowerClip позволяет сообщение в строке состояния, например: (Эллипс PowerClip вкл. Слой 1).

Доступ к операциям PowerClip возможен из меню **Эффекты** ▶ **PowerClip** или в стыкуемой палитре **Диспетчер объектов**. В первом случае необходимо выбрать объект-вложение, дать команду **Эффекты** ▶ **PowerClip** ▶ **Поместить в контейнер** и указать курсором объект-контейнер.

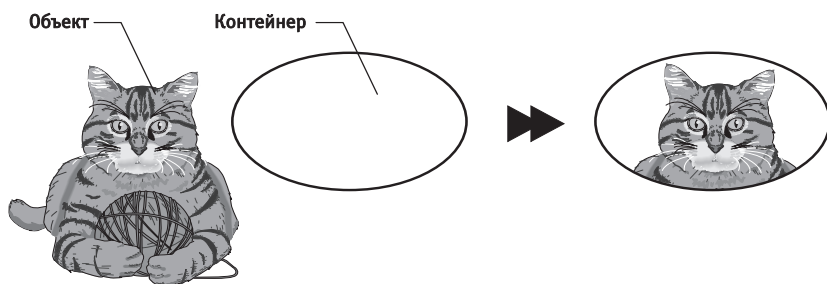


Рис. 16.15. Пример использования контейнера

В стыкуемой палитре **Диспетчер объектов** щелчком правой кнопки мыши необходимо выбрать имя объекта-вложения и перетаскиванием переместить его поверх имени объекта-контейнера. После отпускания кнопки мыши появляется всплывающее меню, в котором надо выбрать пункт **Поместить в контейнер**.

Для редактирования вложенного объекта необходимо выбрать его имя щелчком правой кнопки мыши в стыкуемой палитре **Диспетчер объектов** и в открывшемся всплывающем меню выбрать пункт **Редактировать содержимое**. Редактируемый объект для удобства можно перемещать в пределах рабочего поля. По завершении редактирования следует вновь щелкнуть правой кнопкой мыши на имени объекта и в открывшемся всплывающем меню выбрать пункт **Завершить редактирование**.

Извлечение содержимого контейнера происходит по той же методике, только в меню необходимо выбрать пункт **Извлечь содержимое**. Содержимое контейнера извлекается только целиком.

Пример. Рисование топографической карты

В качестве образца выбрана топографическая карта района юго-западнее Прохоровки, где в июле 1943 года происходили ожесточенные танковые бои между немецкими и советскими войсками. Топографическая съемка района для данной карты проведена в 1946 году и поэтому более адекватно отражает состояние местности в военный период, чем современные карты.

Первый этап

До начала работы необходимо определить размер бумаги для создаваемого документа. Технологически удобным вариантом является размер, совпадающий с размером карты. В этом случае легче переносить в документ данные с оригинала. Однако лист карты превышает по формату А3, что приводит к проблемам при выводе на печать и публикации документа в электронном виде. Более универсальным форматом является А4, поэтому при создании нового документа в программе CorelDraw в раскрывающемся списке **Размер страницы** выбираем строку А4.

Каждая топографическая карта имеет координатную сетку, облегчающую привязку к местности. Для точного позиционирования элементов карты надо воспроизвести такую сетку в документе. Для этого создаем прямоугольник, соответствующий размеру карты, и разбиваем его на нужное число ячеек, например, с помощью инструмента **Разлинованная бумага**.

Важнейшей задачей, решение которой определяет успех всего проекта, является точное воспроизведение карты высот. Основой здесь служат высотные горизонталы, представленные на исходном листе. Создание карты высот возможно различными способами, например:

- ☐ прорисовка высотных горизонталей вручную;
- ☐ трассировка сканированного изображения исходной карты;
- ☐ преобразование сканированного изображения в растровую карту высот в программе *Adobe Photoshop*.

В нашем примере использован метод ручной прорисовки высотных горизонталей как наиболее простой и доступный. Для начала определяем по карте шкалу высот, то есть число высотных горизонталей. В нашем примере высотные горизонталы расположены в диапазоне от 160 до 240 метров с шагом 20 метров. Вспомогательные горизонталы имеют шаг 10 метров.

Вспомним, что в документе CorelDraw все объекты распределены по высоте относительно базового слоя (Слой 1). Таким образом, надо определить такую очередность рисования горизонталей, чтобы предыдущий объект не перекрывал текущий. Этому требованию отвечает метод рисования сверху вниз, от самой высотной горизонтали к нижележащим.

Для рисования выбираем инструмент **Свободная форма**, на панели свойств устанавливаем движок **Сглаживание свободной формы** в положение 75, в раскрывающемся списке **Толщина абриса** выбираем значение 0,176 мм. В палитре цветов щелчком правой кнопкой мыши выбираем цвет контура — отдельно для каждой горизонтали высот (например, 240 м — красный, 220 м — зеленый и т. д.). Последовательно прорисовываем все горизонталы высот. Обращаем внимание, что каждый контур горизонтали высот должен быть замкнутым.

Затем группируем горизонталы одной высоты. Для этого в менеджере объектов проводим выборку кривых одного цвета и даем команду **Упорядочить ► Сгруппировать**. После создания группы выбираем ее в менеджере объектов, щелчком правой кнопкой мыши открываем контекстное меню, выбираем пункт **Переименовать** и вводим «говорящее» имя для группы, например «высота 240».

После создания групп, объединяющих горизонталы одной высоты, присваиваем объектам группы одинаковый цвет заполнения. Для этого удобно воспользоваться

средством Однородная заливка. В диалоговом окне открываем вкладку Краскосмесители. В раскрывающемся списке Модель выбираем цветовую модель, например СМУК (если документ предполагается печатать). В раскрывающемся списке Оттенки выбираем форму представления, например Пятиугольник. В раскрывающемся списке Вариация выбираем вариант градации оттенков цвета, например Светлее. В представленной ниже палитре щелчком мыши выбираем конкретный цвет для заполнения горизонталей высот. На наш взгляд, удобно использовать оттенки зеленого цвета: чем темнее, тем меньше высота. Щелчком на кнопке ОК завершаем операцию.

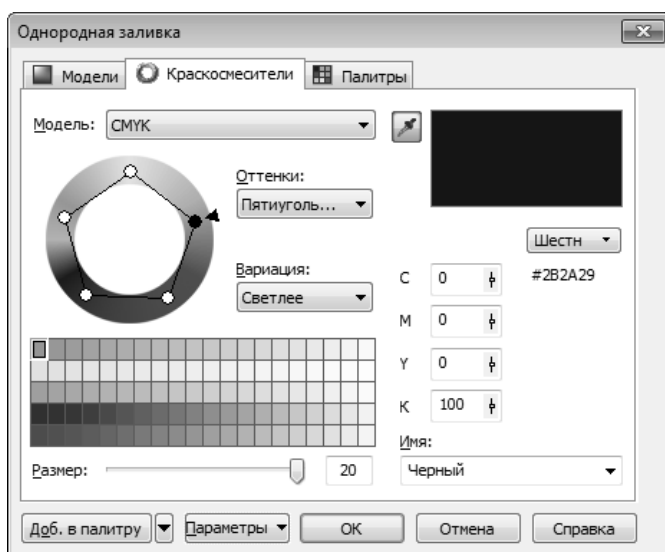


Рис. 16.16. Использование средства Однородная заливка

В завершение первого этапа работы расставляем отметки командных высот, обозначенных на карте. Координаты отметок определяем относительно созданной нами ранее сетки.

Некоторые из нарисованных горизонталей могут выходить за границы карты, поэтому надо «вписать» объекты в установленный обрезной размер. Эту задачу можно решить с помощью инструмента Нож или операцией Вычитание.

Второй этап

На втором этапе работы поверх карты высот накладываются обозначения других элементов местности: реки, водоемы, болота, леса, сады, кустарник и прочие, которые имеют значение в контексте задач, решаемых проектом. Привязку объектов к карте высот осуществляют при помощи координатной сетки.

Реки и водоемы удобно рисовать инструментом Свободная форма с параметром Сглаживание свободной формы около 100. Толщину контура выбирают в соответствии с масштабом изображения на карте. Для водоемов контур должен быть замкнутым. Цвет контура и заливки замкнутых объектов подбирают согласно

методике, описанной выше для карты высот. Созданные объекты данной категории объединяют в группу и присваивают ей имя, например «реки и водоемы».

Леса удобно рисовать инструментом **Кривая Безье**. Сначала проставляют опорные точки по границам лесного массива. Затем с помощью инструмента **Фигура** добавляют при необходимости опорные точки и редактируют форму кривой, а также свойства узлов. Заполнение объектов можно выполнить либо однородным цветом, либо узором, созданным в соответствии с правилами оформления топографических карт. На наш взгляд, в прикладных задачах достаточно выделить растительность различного типа цветом. Например, хвойные леса — более темным оттенком зеленого цвета.

В зависимости от сложности растительного покрова можно сгруппировать однотипные объекты, а затем создать общую группу. Например, в такой последовательности:

- ☐ лиственные леса;
- ☐ хвойные леса;
- ☐ лесозащитные полосы;
- ☐ сады;
- ☐ кустарник;
- ☐ отдельно стоящие деревья;
- ☐ прочие посадки.

Каждая группа должна выделяться цветом или иным типом заполнения. По правилам оформления топографических карт существуют определенные требования к простановке характеристик растительности; например, в лесном массиве указывают средние диаметр ствола, высоту деревьев и расстояние между ними. Если необходимо создать полный аналог топографической карты, придется переносить эти данные с оригинала. Для прикладных целей обычно достаточно указать границы лесных массивов и выделить их цветом.

Третий этап

На третьем этапе работы необходимо нарисовать дорожную сеть. Так как дороги в плане представляют собой сложное сочетание кривых и прямых линий, удобно использовать инструмент **Ломаная линия**. На панели свойств задают толщину контура (**Толщина абриса**) в соответствии с характером дорожного покрытия и масштабом карты.

В прикладных целях дороги различного класса можно выделить разным цветом и толщиной контура. При создании точной карты необходимо следовать установленным правилам оформления и обозначать соответствующими знаками элементы дорожной инфраструктуры: дорожное полотно, мосты, путепроводы, насыпи, выемки, тоннели и пр.

Железнодорожные пути обозначают либо в точном соответствии с правилами оформления топографических карт, либо упрощенно, как прерывистую черно-белую линию. Такую линию можно создать по следующей методике:

- ☐ провести инструментом **Ломаная линия** или любым другим траекторию трассы железнодорожного полотна;

- ❑ установить на панели свойств толщину контура (Толщина абриса) в соответствии с масштабом карты;
- ❑ щелчком правой кнопкой мыши на палитре цветов задать черный цвет контура;
- ❑ создать дубликат контура (CTRL+D или Правка ► Копировать) и задать ему те же координаты, что и оригиналу;
- ❑ щелчком правой кнопкой мыши на палитре цветов задать белый цвет контура дубликата;
- ❑ на панели свойств в поле Толщина абриса задать значение, меньшее на 0,2–0,4 мм исходной толщины (в зависимости от масштаба карты);
- ❑ на панели свойств в раскрывающемся списке Стилль линии указать тот вариант прерывистой линии, который соответствует масштабу карты;
- ❑ сгруппировать оригинал контура и дубликат, присвоить группе имя, например «железная дорога».

К дорожной сети можно отнести и некоторые другие элементы инфраструктуры: линии электропередач, трубопроводы, линии связи и прочие протяженные объекты. Их удобно группировать по типам и целевому назначению.

Четвертый этап

На завершающем этапе работы необходимо нарисовать населенные пункты и другие объекты инфраструктуры, которые остались неохваченными на предыдущих этапах. Также следует установить и реализовать систему текстовых обозначений.

На топографических картах к обозначению населенных пунктов предъявляются высокие требования: на крупномасштабной карте представлено все, вплоть до отдельного сарая или колодца. В прикладных задачах такой точности, как правило, не требуется. В нашем примере мы ограничились обозначением границ населенных пунктов. Приемы рисования контуров и заливки цветом аналогичны тем, которые применялись для рисования лесных массивов. По окончании рисования населенных пунктов желательно объединить эти объекты в отдельную группу и присвоить ей название.

Без пояснительных надписей топографическая карта выглядит малоинформативной. Поэтому необходимо определиться с системой обозначений, прежде всего, с ее полнотой. Оригинал топографической карты содержит сотни обозначений, однако в прикладных целях нам достаточно указать названия населенных пунктов, элементов местности и проставить отметки командных высот.

В качестве шрифта для пояснительных надписей принято использовать начертание без засечек. Для карт разработаны специальные наборы знаков, например *Geographic Symbol*, однако они не содержат знаков алфавита. Шрифт для надписей должен быть компактным, поскольку некоторые названия содержат много букв. В нашем примере использовался шрифт *CompactC*, отличающийся высокой емкостью. На карте обозначены только отметки высот и названия населенных пунктов, что обусловлено малыми размерами листа. Если бы карта создавалась в реальном масштабе 1 : 200 000, то на ней можно было разместить гораздо больше текстовой информации и обозначений. Но карта, перегруженная информацией, читается с трудом и не позволяет выделить ключевые элементы.

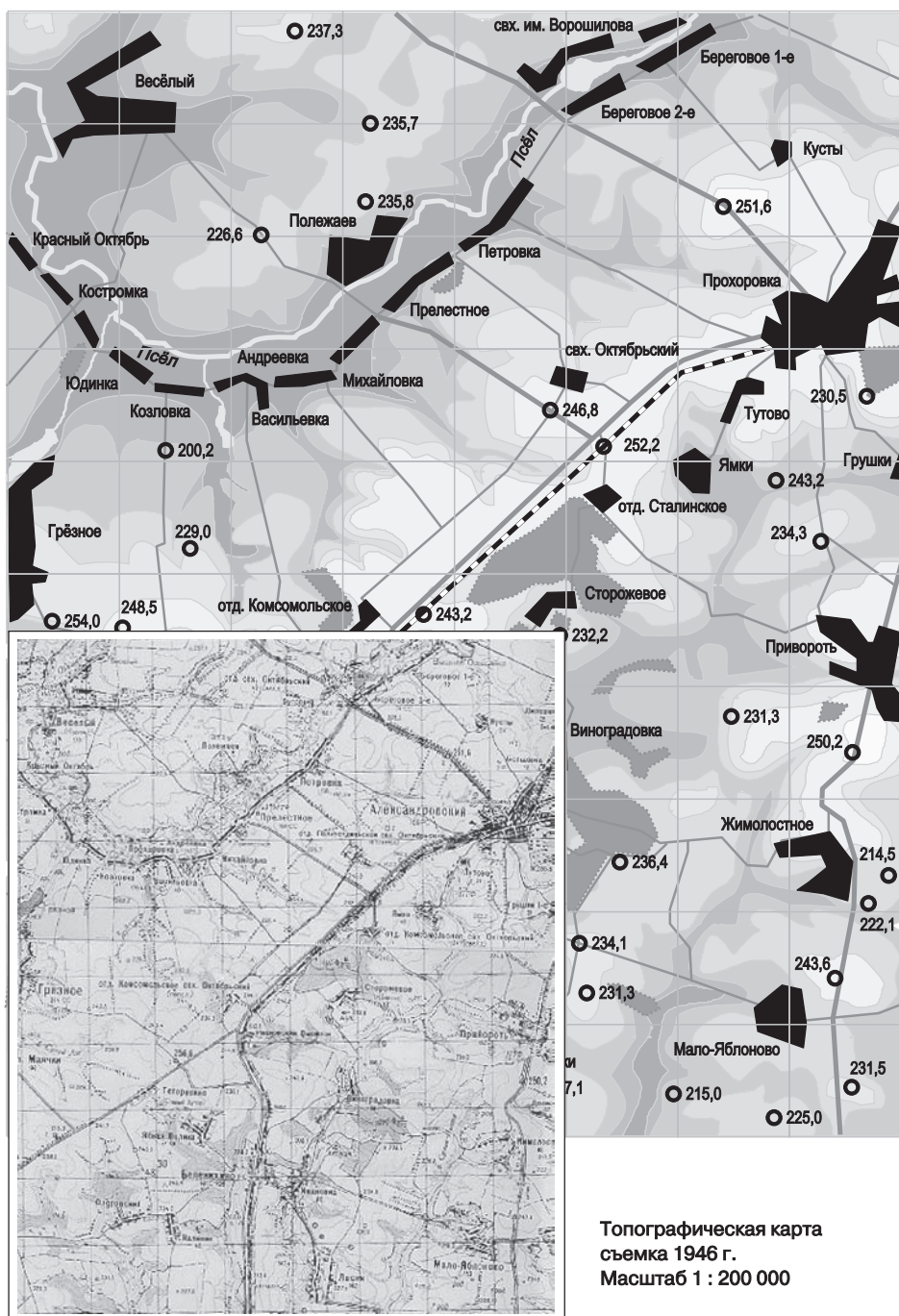
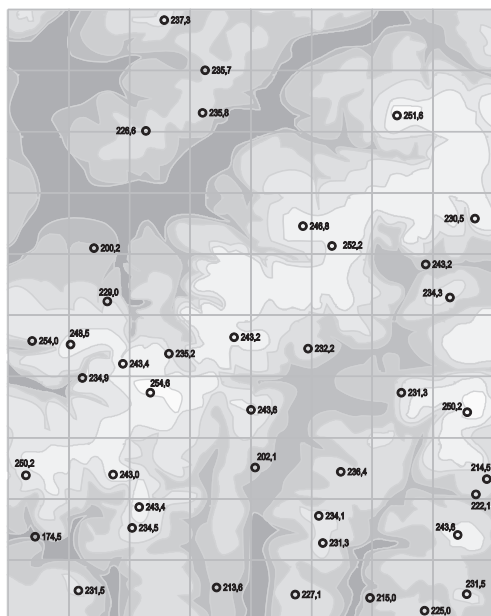


Рис. 16.17. Разработка топографической карты на основе оригинала

Первый этап. Карта высот



Второй этап. Леса и водоемы



Третий этап. Дорожная сеть



Четвертый этап. Населенные пункты и местные предметы



Рис. 16.18. Этапы создания топографической карты

В завершение работы откройте менеджер объектов и проверьте, чтобы взаимное положение имеющихся в проекте групп соответствовало задачам отображения информации. Например, объекты группы «высотные горизонталы» должны находиться на самом нижнем слое в проекте, в противном случае они могут закрыть другие важные элементы.

Практическое занятие

Упражнение 16.1. Настройка CorelDraw

1. Запустите программу CorelDraw (Пуск ► Все программы ► CorelDRAW Graphics Suite X5 ► CorelDraw X5).
2. В открывшемся окне Быстрый запуск щелкните на кнопке Новый пустой документ. В появившемся окне Создание документа оставьте параметры по умолчанию и нажмите ОК.
3. На панели инструментов, которая находится под строкой меню, щелкните на кнопке Параметры.
4. В открывшемся окне Параметры пройдите по пути Рабочее пространство ► Общие ► Число отмен. Установите параметр Обычные операции в значение 30, Растровые эффекты в значение 10. Благодаря такой настройке вы расширите доступное число операций, которое программа позволяет отменить.
5. Раскройте раздел Рабочее пространство ► Документ ► Общие, в раскрывающемся списке режимов просмотра вариант Расширенный. Если при работе в редакторе ощущается торможение, можно выбрать другой, менее качественный, но более быстрый режим просмотра. Выберите раздел Документ ► Сетка. В разделе Настраиваемая сетка установите счетчиками По горизонтали и По вертикали значения 5 мм. Установите флажок Показывать сетку.
6. Выберите раздел Документ ► Стили. В окне, в списке просмотра объектов, для которых назначены стили, выберите строку Графика по умолчанию и щелкните на кнопке Изменить в группе Абрис. В открывшемся окне Перо абриса переключите единицы измерения толщины в пункты и счетчиком Ширина установите значение 1 пункт, щелчком на кнопке ОК закройте окно.
7. В разделе Стили в окне выберите строку Простой текст по умолчанию. На правом поле в разделе настройки шрифтов щелкните на кнопке Изменить. В открывшемся окне Форматировать текст на вкладке Символ в раскрывающемся списке Шрифт выберите шрифт без засечек, например Arial. Счетчиком Размер установите величину кегля 12 пунктов. Перейдите на вкладку Абзац, установите переключатель типа выравнивания По центру. Щелчком на кнопке ОК закройте окно.
8. В разделе Стили в окне выберите строку Фигурный текст и повторите действия, изложенные в пункте 7.
9. В разделе Размер страницы установите в списке Размер параметр A4, установите переключатель Портретная. В раскрывающемся списке Paper (Бумага) выберите формат A4.

10. Перейдите в раздел **Документ**. Установите флажок **Сохранить параметры по умолчанию** для новых документов и все активизировавшиеся после этого флажки.
11. Перейдите в раздел **Рабочее пространство** ▶ **Настройка** ▶ **Команды**. В списке команд найдите команду **Грубая кисть**. В правой части окна выберите вкладку **Сочетания клавиш**, установите курсор в поле **Новое сочетание клавиш** и нажмите **CTRL+ALT+D**. Нажмите на кнопку **Назначить**. Теперь для вызова инструмента **Грубая кисть** вам будет достаточно использовать указанное сочетание клавиш.
12. В окне **Параметры** перейдите по пути **Рабочее пространство** ▶ **Сохранить**. Установите параметр **Автосохранение** в значение **каждые 10 минут**. Благодаря этой установке автоматическое сохранение документа будет производиться чаще, чем это установлено по умолчанию, а значит, в случае системного сбоя, аварийного отключения питания или другой подобной непредвиденной ситуации будет сохранена большая часть вашей работы.
13. Щелчком на кнопке **ОК** закройте окно **Параметры**.

Итог. В этом упражнении мы научились основным приемам настройки интерфейса программы и параметров документа, создали собственное сочетание «горячих» клавиш и элемент управления панели свойств, появляющийся при выборе нескольких объектов.

Упражнение 16.2. Построение графика зависимости крутящего момента двигателя от числа оборотов

1. Запустите программу **CorelDraw** (**Пуск** ▶ **Все программы** ▶ **CorelDRAW Graphics Suite X5** ▶ **CorelDraw X5**).
2. В открывшемся окне **Быстрый запуск** щелкните на кнопке **Новый пустой документ**. В появившемся окне **Создание документа** оставьте параметры по умолчанию и нажмите **ОК**.
3. Выберите инструмент **Разлинованная бумага**, на панели свойств установите значения счетчиков **Столбцы** и **строки** по **20**.
Протягиванием создайте объект с размерами **100×100 мм**.
4. Щелчком правой кнопки мыши в палитре цветов на значении **20% (Серый 20%)** задайте цвет контуров объекта.
5. Выберите инструмент **Кривая Безье**. Щелчками выше левого верхнего угла, в левом нижнем углу, правее правого нижнего угла объекта при нажатой клавише **CTRL** создайте прямые линии, пересекающиеся под прямым углом (для осей координат). На панели свойств выберите с помощью раскрывающихся кнопок **Начальный наконечник** и **Конечный наконечник** окончания линии в виде стрелок.
6. Откройте палитру **Диспетчер объектов** командой **Окно** ▶ **Окна настройки** ▶ **Диспетчер объектов**. Двойным щелчком на имени линии войдите в режим редактирования объекта и измените имя на **Coordinate**. Таким же образом поменяйте имя второго объекта на **Graph**.
7. Инструментом **Выбор** выберите последовательно объекты **Coordinate** и **Graph**. Командой **Упорядочить** ▶ **Выровнять и распределить** ▶ **Выровнять и распределить**

откройте соответствующее окно. Установите флажки По верхнему краю и По левому краю.

8. Выберите инструмент Текст. Щелкните на свободном месте страницы и введите текст «N, Нм». Перетащите текстовый объект к окончанию вертикальной координатной прямой. Таким же образом создайте текстовый объект «v, об/с-1» и перетащите его к окончанию горизонтальной координаты. Изменение шрифта и позиций символов выполните в окнах Форматирование символов, Форматирование абзаца, открываемых из меню Текст.
9. С помощью инструмента Текст поставьте в начале координат цифру 0, а под каждой пятой вертикальной линией объекта Graph числа от 1000 до 4000. Выровняйте эти цифры по вертикали с использованием изученных приемов. Так же расставьте вдоль вертикальной координаты числа от 100 до 400 и выровняйте их по горизонтали.
10. Выберите инструмент Кривая Безье и щелчками мыши в точках с координатами x, y (800, 40; 1200, 80; 1600, 200; 2000, 300; 2600, 400; 3000, 360; 3600, 320) создайте линию. Эти точки являются точками отсчета. Если в вашем распоряжении имеются данные внешней характеристики другого двигателя, можно наложить их для дальнейшего сравнительного анализа.
11. Выберите инструмент Фигура, выберите вторую опорную точку линии, щелчком правой кнопки мыши откройте контекстное меню и щелкните на кнопке Преобразовать в кривую. Подобным образом преобразуйте остальные точки, кроме первой.
12. Выберите предпоследнюю опорную точку кривой, щелчком правой кнопки мыши откройте контекстное меню, щелкните на строке Сглаживать. Подобным образом преобразуйте остальные точки, кроме первой и последней.

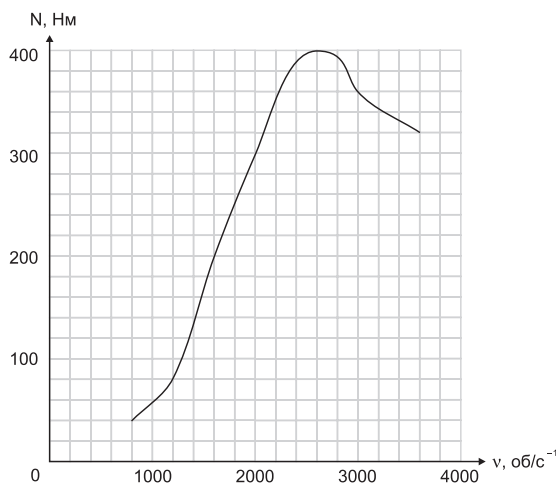


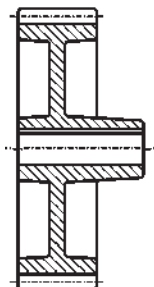
Рис. 16.19. Вариант оформления графика

13. Выберите самую верхнюю точку (точку максимума), щелчком правой кнопки мыши откройте контекстное меню, щелкните на строке **Симметрическая**. Затем, воздействуя на управляющие касательные, добейтесь того, чтобы узловая точка совпала с верхней точкой графика. Итоговый график должен соответствовать представленному на рис. 16.19.

Итог. Мы научились средствами CorelDraw создавать сложные графическо-текстовые документы для иллюстрирования физических процессов.

Упражнение 16.3. Построение упрощенного чертежа разреза зубчатого колеса

1. Запустите программу CorelDraw (Пуск ► Все программы ► CorelDRAW Graphics Suite X5 ► CorelDraw X5).



2. В открывшемся окне **Быстрый запуск** щелкните на кнопке **Новый пустой документ**. В появившемся окне **Создание документа** оставьте параметры по умолчанию и нажмите **ОК**.
3. Чертеж требует высокой точности графики. Поэтому необходимо поменять некоторые установки параметров документа. Щелчком на кнопке **Параметры** панели инструментов, расположенной под строкой меню, откройте диалоговое окно **Параметры**. Откройте раздел **Документ** и перейдите к строке **Сетка**. На вкладке **Сетка** установите в счетчиках **По горизонтали** и **По вертикали** значение 10. Установите флажки **Показывать сетку** и **Привязывать к сетке**. Перейдите к разделу **Направляющие**, установите флажок **Привязывать к направляющим**. Щелчком на кнопке **ОК** закройте окно.
4. Для задания габаритов чертежа следует создать направляющие линии. Установите указатель мыши на вертикальную измерительную линейку и протягиванием перетащите направляющую линию в область чертежа. На панели свойств счетчиком x выставьте значение 20 мм. Следующие линии выставьте в положения $x = 70$ мм и $x = 100$ мм. Вертикальные направляющие выставьте в положения $y = 250$ мм, $y = 160$ мм, $y = 70$ мм.
5. Прежде чем начинать рисование, надо задать толщину контуров, принятую по умолчанию. Щелчком на кнопке **Перо абриса** на Панели инструментов откройте диалоговое окно **Перо абриса**, установите флажок **Графика**, щелчком на

кнопке ОК закройте окно, в открывшемся диалоговом окне **Перо абриса** в разделе **Ширина** в раскрывающемся списке единиц измерения выберите пункт **миллиметры**, счетчиком **Ширина** выставьте значение 0,6. Щелчком на кнопке ОК закройте окно.

Выберите инструмент **Прямоугольник**, нарисуйте прямоугольник размером 50 × 90 мм с координатами центра x, y (45; 115). Откройте палитру **Диспетчер объектов** командой **Окно ▶ Окна настройки ▶ Диспетчер объектов**, присвойте прямоугольнику имя 01. Щелчком правой кнопки мыши на палитре цветов присвойте контуру цвет **Красный**.

6. На панели свойств щелчком на кнопке **Изменить углы совместно** отключите режим взаимосвязанного изменения радиусов закругления углов. Счетчиками **Радиус угла (для левых углов)** и **Радиус угла (для правых углов)** выставьте значение 10 для нижних углов.

Нарисуйте прямоугольник (см. п. 6) размером 50 × 70 мм с координатами центра x, y (45; 115). Присвойте ему имя 02.

Нарисуйте прямоугольник размером 20 × 50 мм с координатами центра x, y (30; 115). Присвойте ему имя 03. Выставьте значение скругления его правых углов 20. Дав команду **Эффекты ▶ Добавить перспективу**, измените перспективу перетаскиванием нижней правой управляющей точки (при нажатом сочетании клавиш **CTRL+SHIFT**) по вертикали до появления в строке состояния **Точка схода по горизонту** значения 461 мм.

7. Создайте дубликат прямоугольника 03 (**CTRL+D**), зеркально отразите его по горизонтали, установите координаты центра x, y (60; 115), присвойте объекту имя 04.

Нарисуйте прямоугольник размером 30 × 10 мм с координатами центра x, y (85; 145), выставьте значение скругления правого нижнего угла 60, присвойте ему имя 05. Дав команду **Эффекты ▶ Добавить перспективу**, измените перспективу перетаскиванием нижней правой управляющей точки по вертикали до появления в строке состояния значения **Точка схода по горизонту** примерно 255 мм.

8. Откройте палитру **Формирование** командой **Окно ▶ Окна настройки ▶ Формирование**. Выберите режим **Объединение**. Выберите прямоугольник 05, **Объединить** и укажите прямоугольник 02. Назначьте контуру итогового объекта цвет **Зеленый**, оставьте имя 02.
9. Выберите прямоугольник 04, в стыкуемой палитре **Преобразование** переключитесь в режим **Пересечение**, сбросьте все флажки, щелкните на кнопке **Пересечь**, укажите объект 02.
10. Повторите операцию (см. п. 13) с прямоугольником 03.
11. Выделите все объекты, сделайте дубликат, отразите его зеркально по вертикали, переместите дубликат в положение x, y (60; 205).
12. Выберите прямоугольник 01, дайте команду **Объединить** и укажите дубликат прямоугольника 01. Присвойте дубликату объекта 02 имя 02а.

Нарисуйте прямоугольник размером 80×20 мм с координатами центра x, y (60; 160), присвойте ему имя 06, назначьте цвет контура Голубой.

Нарисуйте прямоугольник размером 80×30 мм с координатами центра x, y (60; 171,4), присвойте ему имя 07, назначьте цвет контура Голубой.

13. Последовательно выберите прямоугольники 06 и 07 и скомбинируйте их. Присвойте заполнению цвет Белый. Присвойте итоговому объекту имя 08.
14. Выберите объекты 02 и 02а, продублируйте их, присвойте заполнению цвет White (Белый), переместите на задний план командой Упорядочить ► Порядок ► На задний план страницы, переместите их в положение x, y (60; 160), сгруппируйте объекты, присвойте результату имя Group 01. Выберите объект 01, переместите его на задний план. Сгруппируйте объекты 02 и 02а, присвойте результату имя Group 02. Командами Упорядочить ► Порядок разместите объекты в следующей последовательности (сверху вниз): 08, Group 02, Group 01, 01.
15. Выберите объект Group 01, щелчком на кнопке Заливка узором PostScript откройте окно Текстура PostScript. Выберите строку Штриховка, счетчиками установите параметры: Макс.расстояние = 50, Мин. расстояние = 50, Ширина линии = 5, Угол = 45, Случайный отсчет = 0.
16. Инструментом Кривая Безье проведите горизонтальную линию длиной 100 мм с координатами x, y (60, 160). На панели свойств установите ее свойства: Толщина абриса = 0,353 мм; Стиль линии — штрих-пунктир.
17. Инструментом Кривая Безье проведите горизонтальную линию длиной 60 мм с координатами x, y (45, 245). На панели свойств задайте ее свойства: Толщина абриса = 0,353 мм; Стиль линии — штрих-пунктир. Создайте такую же линию с координатами x, y (45; 75).
18. Выберите все объекты, присвойте контурам цвет Черный, сгруппируйте объекты. При необходимости сохраните чертеж под именем, указанным инструментом.

Итог. Мы изучили ряд приемов использования средств CorelDraw для построения точного чертежа машиностроительной детали.

Глава семнадцатая

Автоматизация обработки документов

Компьютер предназначен для работы с документами, имеющими электронную форму. В то же время нам часто приходится иметь дело с бумажными изданиями и документами: журналами, книгами, письмами, служебными записками и т. д. Чтобы в работе с информацией такого рода тоже можно было использовать компьютер, необходимы средства *преобразования бумажных документов в электронную форму*.

Если предполагается, что документ содержит в основном текстовую информацию, то можно выделить следующие основные этапы такого преобразования:

- ❑ в ходе *сканирования* при помощи устройств оцифровки изображения производится создание электронного образа (изображения) документа;
- ❑ процесс *распознавания* позволяет преобразовать электронное изображение в текстовые данные (с сохранением элементов форматирования оригинала или без них);
- ❑ для документов, исполненных на иностранном языке, применяют дополнительные средства *автоматизированного перевода на другой язык*.

17.1. Преобразование документов в электронную форму

Сканирование документов

Процесс создания электронного изображения бумажного документа напоминает его фотографирование и требует применения соответствующего устройства. Сегодня в качестве такого устройства выступает *сканер*. Современные *цифровые камеры* также могут обеспечить для документов стандартного формата качество изображения, достаточное для их надежного распознавания. Однако применять сканеры, в частности планшетные, гораздо удобнее.

Основной рабочий элемент сканера включает источник света, используемый для освещения документа, и светочувствительную головку, воспринимающую отраженный свет. Универсальные сканеры, в отношении которых нет специальных требований по функциональным возможностям, качеству и скорости сканирования, делятся на три основные категории.

Ручной сканер протягивается над поверхностью документа вручную. Он обеспечивает минимальное качество сканирования, в частности непригоден для сканирования документов, содержащих иллюстрации. Применяется обычно в условиях, когда использование настольных устройств сканирования невозможно или неудобно.



Листовой сканер способен сканировать отдельные страницы, протягивая их мимо светочувствительного элемента. Его недостатком является невозможность сканирования книг и журналов без разборки на

отдельные страницы. В связи со значительным снижением стоимости планшетных сканеров практически вышел из употребления.



В *планшетном сканере* подвижный светочувствительный элемент перемещается в ходе сканирования внутри корпуса устройства. Сканируемый документ располагается напротив прозрачного окна в корпусе прибора. Этот вид сканера лишен недостатков, которые характерны для типов, рассмотренных выше. Сегодня планшетный сканер является наиболее распространенным типом сканера.



Сканер является внешним устройством и подключается к компьютеру через специальный разъем. При высоком разрешении и большой площади сканируемого документа объем передаваемых данных оказывается очень большим и требует производительной линии передачи. Современные сканеры чаще всего используют для подключения порт USB. Сканеры предыдущих поколений использовали порт принтера (обеспечивает низкую производительность) или интерфейс *SCSI (Small Computer System Interface)*.

Разные модели сканеров понимают разные управляющие команды. Чтобы избежать разнобоя, был принят универсальный стандарт взаимодействия сканера и приложений. Этот стандарт называется *TWAIN*. Приложение посылает команды *драйверу TWAIN*, который преобразует их в инструкции, распознаваемые конкретным сканером. Таким образом, для приложения конкретная модель сканера перестает иметь значение. Начиная с Windows ME, операционные системы от Microsoft поддерживают интерфейс WIA. Современные сканеры могут работать как с WIA, так и с TWAIN.

Операционная система Windows 7 также поддерживает интерфейс TWAIN.

Сканирование через посредство интерфейса TWAIN осуществляется следующим образом. Сначала следует включить сканер. Команда сканирования располагается в приложении в меню *Файл* (например, в программе Paint соответствующий пункт называется *Со сканера или камеры*). После выбора этой команды открывается диалоговое окно драйвера TWAIN, вид которого зависит от модели сканера

(рис. 17.1). В этом окне задают параметры сканирования: черно-белый или цветной режим, разрешение, коррекция яркости и контрастности. Большинство сканеров позволяют также произвести предварительное черновое сканирование с низким разрешением и по его результатам точно задать область сканирования — часть страницы документа.

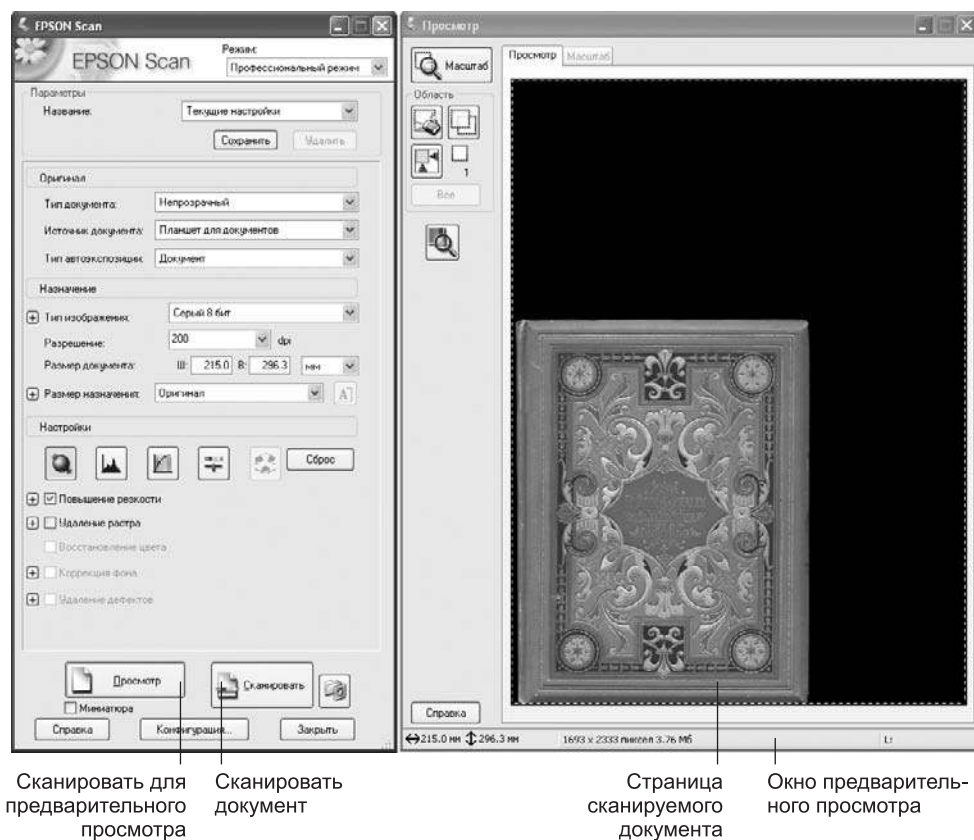


Рис. 17.1. Диалоговое окно интерфейса TWAIN может выглядеть по-разному — его вид определяется драйвером сканера

После настройки всех параметров следует щелкнуть на кнопке **Сканировать** (надпись на кнопке может быть иной). Процесс сканирования происходит автоматически, и изображение передается в приложение. Диалоговое окно драйвера автоматически не закрывается, так что, например, в многооконных графических редакторах (таких, как Adobe PhotoShop) можно сразу провести сканирование нескольких изображений.

Распознавание документов

Этап *распознавания* документа состоит в преобразовании электронного изображения (фактически набора цветных или черно-белых точек) в текстовый документ. Ранее для описания этого процесса обычно использовался термин *OCR (Optical Character Recognition)*, который соответствует одному из используемых методов. При таком подходе происходит «сравнение» элемента изображения с эталонными вариантами начертания символов, после чего выбирается наиболее подходящий символ. Этот подход требует использования специального комплекта шрифтов, но дает на нем наилучшие результаты. Современные алгоритмы распознавания не привязаны к конкретному начертанию символов: программа, как и человек, способна узнавать буквы при любых начертаниях (а иногда и при значительных искажениях).

В ходе распознавания сначала в изображении выделяются крупные элементы текста: колонки, абзацы, отдельные текстовые блоки (например, подрисуночные подписи), ячейки таблиц. Этот этап называют *сегментацией*, он может выполняться автоматически или вручную. После этого выполняется автоматический этап распознавания: блоки разбиваются на строки, строки — на отдельные символы, каждый из которых распознается независимо и помещается в итоговый текстовый документ.

Работа с программой FineReader

Все операции, необходимые в ходе преобразования бумажного документа в электронную форму, могут быть выполнены с помощью программы FineReader (рис. 17.2). Эта программа способна выполнять сканирование и распознавание текстов на разных языках, в том числе и смешанных двуязычных текстов. С ее помощью можно выполнять пакетную обработку многостраничных документов, а также настраивать режим распознавания для улучшения соответствия электронного документа бумажному оригиналу при плохом качестве последнего или использовании в нем шрифтов, далеких от стандартных.

Основные операции обработки бумажного документа в программе FineReader выполняются с помощью панели инструментов Главная. С точки зрения этой программы, процесс обработки документа состоит из пяти этапов:

- ☐ сканирование документа (кнопка **Сканировать**) или открытие для распознавания ранее отсканированного документа, сохраненного в виде графического файла (кнопка **Открыть**);
- ☐ автоматическая или ручная (элементы управления в области **Изображение**) сегментация документа;
- ☐ распознавание документа (кнопка **Распознать**);
- ☐ редактирование и проверка результата (кнопки в верхней части области **Текст**);
- ☐ сохранение документа (кнопка **Сохранить**).

Сканирование документа. На этапе *сканирования* производится получение изображений при помощи сканера и сохранение их в виде, удобном для последу-

ющей обработки. Чтобы начать сканирование, надо включить сканер и щелкнуть на кнопке **Сканировать** на панели инструментов. В программе FineReader сканирование может производиться как через драйвер TWAIN, так и в обход него. Первый способ используют, когда требуется точная настройка параметров сканирования, когда документ включает цветные иллюстрации, которые необходимо сохранить, а также когда разные страницы многостраничного документа сильно различаются по качеству. Второй вариант обеспечивает максимальную скорость и удобство сканирования. Выбор используемого варианта осуществляется при помощи переключателя **Использовать интерфейс ABBYY FineReader/Использовать интерфейс сканера** (Сервис ► Опции ► Сканировать/Открыть).

Процесс сканирования осуществляется автоматически и требует от пользователя только вспомогательных операций, таких как смена сканируемой страницы. Возможность вмешательства в работу программы заблокирована размещением на экране специального диалогового окна, уведомляющего о том, что идет сканирование, и позволяющего прервать это процесс.

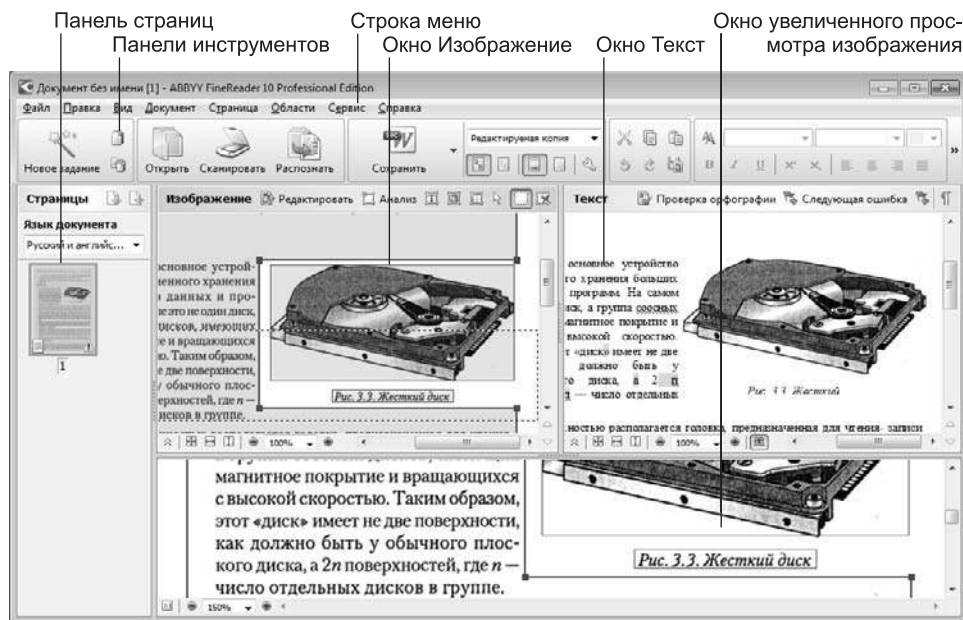


Рис. 17.2. Рабочее окно программы FineReader в процессе распознавания отсканированного документа

По завершении сканирования значки всех обработанных страниц отображаются в окне **Страницы**. В основной части рабочей области появляется окно **Изображение**, содержащее изображение текущей страницы. Добавлять новые изображения в набор страниц можно не только путем сканирования, но и путем открытия файлов с изображениями, имеющихся на компьютере.

Распознавание документа. Этап распознавания текста документа можно разделить на две операции, выполняемые поочередно. Первая операция — сегментация,

разбиение страницы на блоки текста. Обычная книжная страница представляет собой единый блок текста, который распознается в естественном порядке — по строкам, расположенным на странице сверху вниз и идущим от левого края до правого. Если страница содержит колонки, иллюстрации, врезки, подрисовочные подписи или таблицы, то порядок распознавания требует коррекции.

В таком случае содержимое страницы разбивается на *блоки*. Внутри каждого блока сохраняется естественный порядок распознавания. Блоки нумеруются, исходя из порядка включения их в документ. Программа FineReader способна выполнить сегментацию (определить границы блоков) автоматически даже для весьма сложных страниц. При этом учитываются поля документа, просветы между колонками, рамки. По умолчанию сегментация выполняется автоматически перед процессом распознавания текста.

Если структура страницы очень сложна или содержит явные дефекты, которых не всегда удастся избежать при сканировании, удобнее использовать *ручную сегментацию* или ручное редактирование результатов автоматической сегментации. Блоки отображаются в виде цветных прямоугольников. Новый блок создают протягиванием мыши по диагонали прямоугольника. Текущий блок помечается выделенной линией, а его углы — прямоугольными маркерами. С помощью этих маркеров можно изменить размер или положение блока.

Команды для редактирования блоков и редактирования графического представления страницы выведены в верхнюю часть панели Изображение. Они позволяют задавать на странице разные типы блоков, которые обрабатываются программой по-разному. Чтобы изменить тип блока, надо щелкнуть правой кнопкой мыши в его пределах и назначить новый тип с помощью меню Изменить тип области в контекстном меню. Программа FineReader поддерживает следующие типы блоков:

- ☐ автоматический (Зона распознавания) — на этапе распознавания анализируется и разбивается на блоки автоматически;
- ☐ текстовый (Текст) — на этапе распознавания преобразуется в текст;
- ☐ табличный (Таблица) — представляет собой набор ячеек, каждая из которых преобразуется в текст по отдельности;
- ☐ изображение (Картинка) — включается в документ без изменений как графическая иллюстрация, если формат сохранения преобразованного документа допускает вставные объекты;
- ☐ штрих-код (вариант Штрих-код из меню Изменить тип области).

После того как блок размещен на странице, при наведении на него указателя мыши появляется дополнительное меню, содержащее кнопки для управления свойствами блока. В частности, для табличных блоков доступны операции добавления и удаления вертикальных и горизонтальных разделителей, для изображений и текстовых блоков — добавление и удаление частей областей.

Процесс распознавания текста после сегментации полностью автоматизирован. Кнопка Распознать на панели инструментов позволяет автоматически выполнить сегментацию и распознавание, хотя в ходе сегментации учитываются ограничения, установленные вручную. В ходе процесса отображается диалоговое окно Распо-

знание, позволяющее прервать процесс. Кроме того, в этом окне отображаются сообщения, указывающие на наличие проблем при распознавании. Проблемы обычно вызываются неверными настройками или плохим качеством распознаваемого изображения. Если же дело в каких-то шрифтовых особенностях распознаваемого документа, применяют *распознавание с обучением*.

Распознавание с обучением. Распознавание с обучением состоит в формировании *эталона*, который используется в ходе распознавания в дальнейшем. Эталон настраивается так, чтобы соответствовать определенному документу или группе однотипных документов. Чтобы создать эталон, используют команду Сервис ► Редактор эталонов ► Новый. После этого надо указать имя эталона и щелкнуть на кнопке ОК. Режим распознавания с обучением включается при настройке параметров работы программы (Сервис ► Опции ► Распознать). Создать, выбрать или сменить эталон можно, щелкнув на кнопке Эталоны. После этого необходимо установить переключатель Распознавание с обучением.

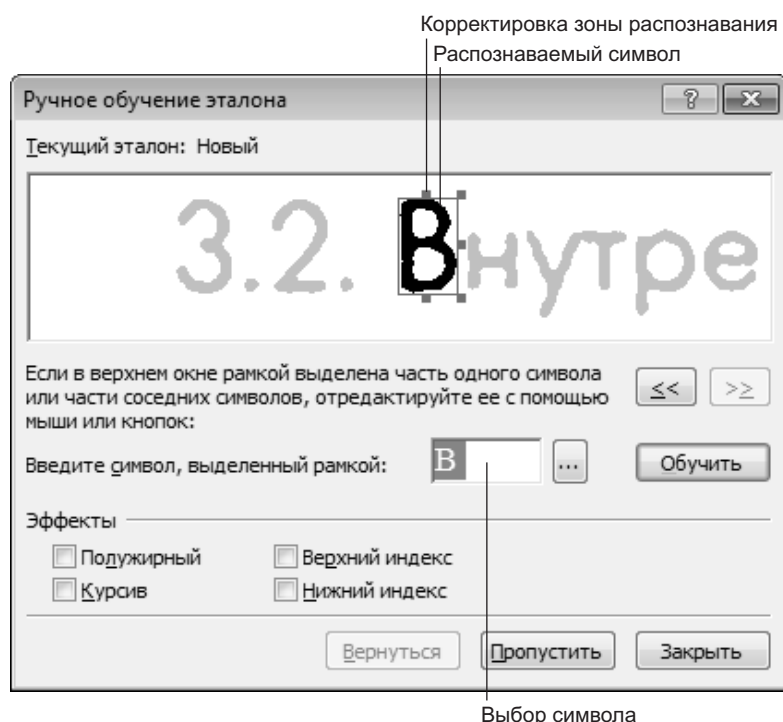


Рис. 17.3. Ручное «обучение» механизма распознавания текста

Когда в ходе распознавания с обучением программа FineReader обнаруживает символ, который не может интерпретировать однозначно, на экран выдается диалоговое окно *Ручное обучение эталона* (рис. 17.3). Программа указывает элемент изображения, вызвавший сомнения, и показывает, как именно он будет интерпретирован. Если допущена ошибка, можно указать нужный символ в поле Введите символ, выделенный рамкой или уточнить область распознавания с помощью кнопок Сдвинуть влево и Сдвинуть вправо. Затем надо щелкнуть на кнопке Обучить.

Необходимые сведения сохраняются и используются при дальнейшем анализе изображения. Если число ошибок невелико, можно продолжить распознавание в обычном режиме щелчком на кнопке **Заккрыть**.

Редактирование документа. Когда распознавание данной страницы завершается, полученный текстовый документ отображается в окне **Текст**. Заключительные этапы работы позволяют отредактировать полученный текст с помощью средств, напоминающих текстовый редактор WordPad (панель для форматирования отображается в правой части панели инструментов, над окном **Текст**). Провести проверку орфографии с учетом трудностей распознавания позволяет кнопка **Проверка орфографии** на панели инструментов окна **Текст**. Информация о возможных ошибках и неоднозначностях отображается в диалоговом окне **Проверка** (рис. 17.4).

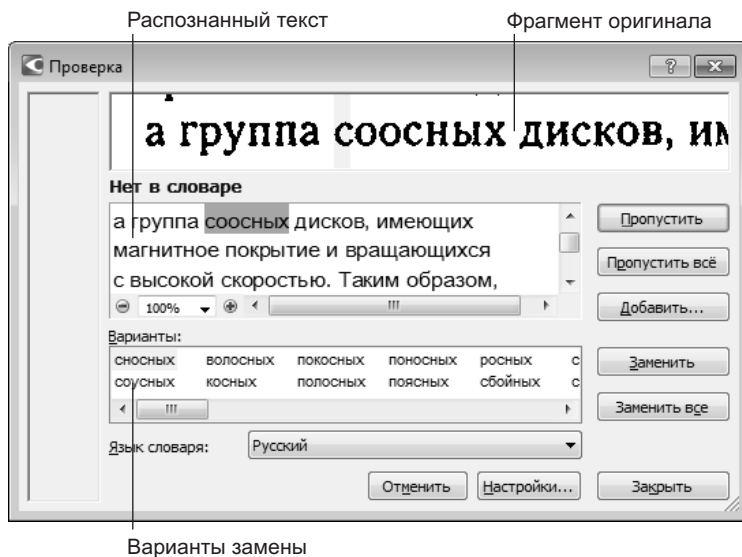


Рис. 17.4. Проверка правильности текста с учетом трудностей распознавания и правил правописания

Сохранение документа. По щелчку на кнопке **Сохранить** на панели инструментов запускается инструмент **Сохранить как**. Он позволяет сохранить распознанный текст или передать его в другую программу (например, в *Microsoft Word*) для последующей обработки.

Практическое занятие

Упражнение 17.1. Сканирование документа

1. Включите сканер.
2. Запустите программу Paint (Пуск ► Все программы ► Стандартные ► Paint).
3. Откройте крышку сканера, положите документ на окно сканера текстом вниз, закройте крышку.
4. Дайте команду Paint ► Со сканера или камеры.

5. Средствами открывшегося диалогового окна драйвера TWAIN проведите предварительное сканирование документа.
6. Средствами диалогового окна драйвера TWAIN выделите на документе область, подлежащую сканированию.
7. Средствами диалогового окна драйвера TWAIN задайте черно-белый режим и разрешение сканирования. Для обычных текстовых документов наиболее эффективное разрешение составляет 300 dpi.
8. Средствами диалогового окна драйвера TWAIN проведите сканирование.
9. Закройте диалоговое окно драйвера TWAIN.
10. Ознакомьтесь с тем, как выглядит отсканированный документ. Увеличьте масштаб изображения, чтобы оценить качество воспроизведения отдельных символов.
11. Сохраните отсканированный документ в формате TIFF для использования в следующем упражнении.

Итог. Мы научились выполнять сканирование документов из приложений через драйвер TWAIN. Это наиболее универсальный метод, применимый во всех приложениях, которые допускают использование сканера.

Упражнение 17.2. Преобразование изображения в текстовый документ

1. Включите сканер.
2. Запустите программу FineReader (Пуск ► Все программы ► ABBYY FineReader 10 ► ABBYY FineReader 10 Professional Edition).
3. Откройте крышку сканера, положите документ на окно сканера текстом вниз, закройте крышку сканера.
4. Щелкните на кнопке **Сканировать** на панели инструментов.
5. Дождитесь окончания сканирования. Обратите внимание на появление значка отсканированного документа на панели **Страницы** и в окне **Изображение**.
6. Щелкните на кнопке **Распознать**. Изучите результат автоматической сегментации в окне **Изображение**. Ознакомьтесь с распознанным текстом в окне **Текст**.
7. Сохраните распознанный текст в виде текстового файла.
8. Откройте текстовый файл в программе Блокнот и еще раз убедитесь в правильности распознавания. Закройте программу Блокнот.
9. Дайте команду **Файл ► Открыть PDF/изображение** и выберите изображение, созданное в предыдущем упражнении.
10. Выберите это изображение в окне **Страницы** и проведите его распознавание в соответствии с пп. 6–8 данного упражнения.
11. Сравните результаты распознавания при сканировании через TWAIN-драйвер и в обход него. Сравните трудоемкость этих операций.

Итог. Мы научились преобразовывать бумажные документы в электронную форму при помощи программы FineReader. Мы познакомились с возможностью работы с пакетом страниц. Мы также выяснили преимущества методов сканирования с использованием TWAIN-драйвера и в обход него.

Упражнение 17.3. Ручная сегментация изображения

1. Включите сканер.
2. Запустите программу FineReader (Пуск ► Все программы ► ABBYY FineReader 10 ► ABBYY FineReader 10 Professional Edition).
3. Откройте крышку сканера, положите документ на окно сканера текстом вниз, закройте крышку.
4. Щелкните на кнопке Сканировать на панели инструментов и дождитесь окончания сканирования.
5. Щелкните на кнопке Распознать на панели инструментов. Ознакомьтесь с тем, как проведено упорядочение распознанного текста в соответствии с автоматической сегментацией. Оцените пригодность полученного документа.
6. Сохраните полученный документ для последующего сравнения.
7. Выполните команду Страница ► Удалить все области и текст.
8. Щелкните на кнопке Выделить текстовый блок на панели инструментов Изображение. Сформируйте блоки вручную, выделяя отдельные элементы документа.
9. Чтобы блоки, содержащие иллюстрации, не распознавались как текст, щелкните на каждом из них правой кнопкой мыши и выберите в контекстном меню команду Изменить тип области ► Картинка.
10. Щелкните на кнопке Распознать на панели инструментов. Сохраните полученный документ.
11. Сравните документы, полученные в результате автоматической и ручной сегментации.

Итог. Мы научились производить ручную сегментацию изображения для распознавания и узнали, в каких случаях ее следует производить. Мы также узнали, как задается тип блоков и порядок блоков, чтобы полученный документ наилучшим образом соответствовал оригиналу.

Упражнение 17.4. Корректировка текста отсканированного документа

1. Включите сканер.
2. Запустите программу FineReader (Пуск ► Программы ► ABBYY FineReader 6.0 Professional).
3. Откройте крышку сканера, положите документ на окно сканера текстом вниз, закройте крышку.
4. Дайте команду Файл ► Новое задание.

5. В окне **Новое задание** перейдите на вкладку **Другие** и нажмите кнопку **Сканировать**, чтобы выполнить сканирование документа.
6. Щелкните на кнопке **Распознать**, чтобы программа выполнила сегментацию и распознавание текста автоматически.
7. Щелкните на кнопке **Проверка орфографии** в окне **Текст**, чтобы начать проверку распознанного текста.
8. В диалоговом окне **Проверка** выбирайте необходимые действия для корректировки ошибок распознавания.
9. Обращайте также внимание на ошибки, которые программа FineReader может пропустить: ошибки при распознавании знаков препинания, пропуски строк при низком качестве отсканированного текста, дефекты сканирования, которые не могут быть исправлены автоматически.
10. Сохраните полученный документ.

Итог. Мы научились улучшать качество электронного документа путем полуавтоматической обработки результатов распознавания. Мы также узнали, какие ошибки наиболее характерны для систем распознавания.

17.2. Автоматизированный перевод документов

К средствам автоматизации перевода можно отнести два вида программ: электронные словари и программы перевода. *Электронные словари* представляют собой средства для перевода *отдельных слов*, отображаемых на экране или имеющих в документе. Удобство их использования состоит в возможности немедленно получить перевод неизвестного слова без поиска его в отдельном толстом томе. *Программы перевода* получают на входе текст, выполненный на одном языке, и выдают текст на другом языке, то есть автоматизируют перевод текста.

Электронные словари удобны для профессиональных переводчиков, которые выполняют большую часть работы по переводу вручную. Их также могут использовать лица, в целом знающие иностранный язык, если надо не обеспечить перевод документа, а просто ознакомиться с его содержанием.

Надежный и качественный автоматический перевод документов с одного языка на другой (мы будем говорить в основном о переводе с английского на русский) пока остается недостижимым идеалом. Причин для этого множество, и главная из них состоит в том, что перевод текста не сводится к переводу отдельных лексических единиц. Преодолеть этот барьер современные программы автоматического перевода пока не могут.

Тем не менее современные средства автоматизации перевода достигли того уровня, который позволяет эффективно использовать их на практике. Дело в том, что *технический текст*, в отличие от *художественного*, использует ограниченное число языковых конструкций и более ориентирован на однозначную интерпретацию. Среди используемых лексических единиц встречается большое число *технических терминов*, имеющих совершенно определенный смысл в рамках данной научной или технической дисциплины. Это значительно упрощает процесс перевода и по-

звolyет в отдельных случаях автоматически получать текст, близкий к результату ручного подстрочного перевода.

Программы автоматического перевода имеет смысл использовать для перевода технических текстов в следующих случаях:

- ☐ при абсолютном незнании иностранного языка;
- ☐ при необходимости получить перевод *быстро*, даже ценой снижения его качества (например, это относится к переводу веб-документов);
- ☐ для перевода на иностранный язык (умения читать иноязычные тексты недостаточно, чтобы научиться объясняться на иностранном языке);
- ☐ для быстрого создания первоначального черновика («подстрочника»), используемого в ходе подготовки полноценного перевода.

Работа с программой PROMT 9

Для автоматизированного перевода технических текстов можно, например, использовать разные программы семейства PROMT. Они позволяют переводить документы с английского языка на русский и с русского на английский. Чтобы обеспечить правильный перевод терминов, относящихся к определенной научной дисциплине, используют специализированные словари, в которых для слов, используемых как термины, предлагается в качестве перевода не «обиходное», а специальное значение.

Если необходимость в переводе документов возникает часто и обрабатывать приходится документы разных форматов, удобно использовать один из вариантов пакета PROMT 9. Если необходимость перевода возникает от случая к случаю и приходится иметь дела с неформатированными текстами (например, сообщениями электронной почты) и короткими отрывками, можно воспользоваться более простыми программами, которые входят в семейство Promt или принадлежат к другим пакетам перевода. В этом пособии мы рассмотрим работу с использованием пакета PROMT XT.

Если программа PROMT установлена на компьютере, для ее запуска можно использовать Главное меню (Пуск ► Все программы ► PROMT ► PROMT professional 9.0).

После запуска откроется окно для выбора операции, которую следует выполнить с помощью программы. В частности, нас интересует операция **Запуск приложений**, при выборе которой отображается список доступных вариантов дальнейших действий. Если выбрать среди этих вариантов **Переводчик Promt**, на экране отобразится окно переводчика (рис. 17.5). При выборе других вариантов возможен перевод различных документов в пакетном режиме с предварительным указанием параметров перевода.

Одновременно для обработки может быть открыто несколько документов. Окна документов имеют необычный вид (см. рис. 17.5). Они разбиваются на три отдельные области: две из них предназначены для отображения оригинала текста и сформированного перевода, а третья представляет собой *информационную панель*, предназначенную для вывода информации о переводимом документе и специальных настройках.

Чтобы произвести перевод имеющегося документа с использованием заданных по умолчанию настроек, применяют следующий порядок действий.

1. Сначала необходимо открыть документ на языке оригинала (команда **Файл ► Открыть**). Нужный документ выбирают в диалоговом окне **Выберите документ**. Формат открываемого файла выбирают в раскрывающемся списке **Тип файлов**.
2. После выбора файла появляется диалоговое окно **Конвертировать файл**. В нем можно уточнить реальный формат документа, хранящегося в файле, если он не соответствует типу файла или когда тип файла может соответствовать нескольким разным форматам документа. В этом же окне можно задать тематику файла, которая определяет особенности перевода.

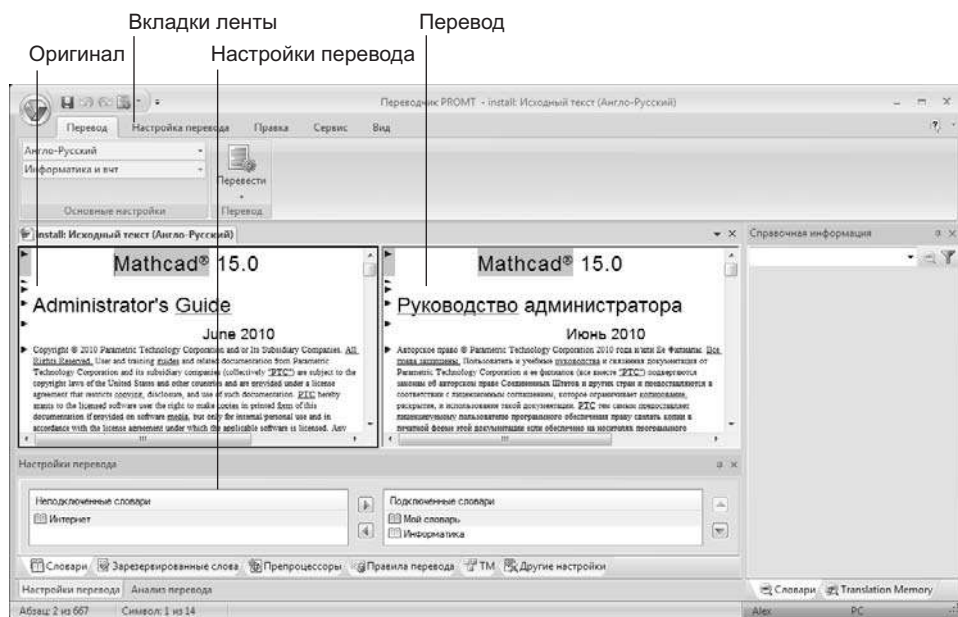


Рис. 17.5. Рабочее окно системы автоматизированного перевода PROMT

3. Документ загружается и отображается в области исходного текста. При вертикальном разбиении окна документа эта область располагается слева. Если предполагается длительная работа над переводом текста, его сохраняют как документ программы *PROMT* (файл с расширением *.STD*).
4. Определение языков оригинала и перевода рассматривается как *направление перевода*. Чтобы выбрать направление перевода, используют поле **Направление перевода** на вкладке **Перевод**.
5. Чтобы перевести весь текст целиком, используют кнопку **Перевести** на вкладке **Перевод**. В ходе выполнения перевода на экране отображается диалоговое окно **Перевод текста** с индикатором хода работы. Перевод текста помещается (при вертикальном разбиении окна) в правую область. Для удобства последующего редактирования перевод снабжается цветовой разметкой: неизвестные программе слова подчеркиваются красным, а зарезервированные слова, которые не надо

переводить, — зеленым. Другие типы маркировки и подчеркивания показывают, из какого словаря взят перевод указанного слова.

Текст, помещенный в областях окна программы PROMT XT, можно редактировать (и оригинал, и перевод). Чтобы заново перевести отредактированный абзац, используют команду Перевести ► Текущий абзац, расположенную в выпадающем меню кнопки Перевести. Текущий абзац — это абзац, в котором располагается текстовый курсор. Он выделяется голубой полосой вдоль левого края.

6. После того как работа с документом в программе PROMT завершена, его сохраняют в одном из общепринятых форматов. Для сохранения только оригинала (возможно, отредактированного) служит команда Файл ► Сохранить ► Исходный текст. Чтобы сохранить переведенный текст, применяют команду Файл ► Сохранить ► Перевод. В обоих случаях можно сохранять как содержимое документа, так и его элементы форматирования, сохраняющие, по возможности, оформление оригинала.

Чтобы продолжить работу с текстом позднее, удобнее сохранить его двуязычный вариант, так называемую *билингву* (Файл ► Сохранить ► Билингву как список абзацев или Билингву как таблицу). В первом случае сохраняется сплошной поток текста, причем абзацы оригинала и перевода чередуются. Во втором случае документ сохраняется как таблица с двумя столбцами. Каждая ячейка содержит один абзац, причем в левом столбце отображается оригинал, а в правом — соответствующий перевод.

Контроль качества перевода

Качество перевода определяется полнотой используемых словарей и учетом грамматических правил. При переводе можно как применять стандартные ресурсы программы, так и добавлять собственные.

Работа со словарями. Правила перевода отдельных слов (терминов) определяются использованием *словарей*. Для каждого переводимого документа задается *набор* применяемых словарей. Словари просматриваются в определенном порядке, и как только переводимое слово обнаружено в каком-то из словарей, дальнейший просмотр прекращается. Программа PROMT использует при переводе три типа словарей.

- ❑ *Генеральный* словарь содержит общеупотребительную лексику и бытовые значения слов. Он используется всегда и притом *самым последним*, если слово не найдено ни в одном из других словарей. Изменение этого словаря невозможно.
- ❑ *Специализированные* словари содержат термины из различных областей знаний, причем значение переводимого термина выбирается в соответствии со специализацией словаря. Одни и те же слова могут иметь совершенно разный смысл в разных технических дисциплинах, так что выбор нужного словаря обеспечивает правильное использование специальной терминологии в переводе. Редактирование специализированных словарей не допускается, но их можно подключать или отключать при переводе документа.
- ❑ *Пользовательский* словарь формируется пользователем вручную. В него можно включить слова, отсутствующие в других словарях, или представить более

точный перевод каких-то из терминов. Пользовательские словари можно произвольно создавать и редактировать. Применяют пользовательские словари обычно в первую очередь, до специализированных и генерального.

Узнать, какие словари используются при переводе, можно в окне Подключенные словари на вкладке Словари информационной панели. Порядок перечисленных словарей соответствует порядку их использования. Генеральный словарь в этом списке не указывается. Чтобы изменить порядок использования словарей, следует щелкнуть на словаре, положение которого в списке надо изменить, правой кнопкой мыши и выбрать в контекстном меню команду Вверх или Вниз.

Транслитерация и резервирование. Не все слова требуют перевода. Обычно без изменений оставляют имена собственные. Иногда при этом используют *транслитерацию* — запись, использующую другой алфавит, но соответствующую написанию или произношению слова на исходном языке. В частности, транслитерация повсеместно используется при передаче иностранных имен и фамилий. Транслитерация не считается переводом.

Иногда необходимо отказаться от перевода целых абзацев. Например, нелепый результат даст попытка перевода исходных текстов программ. То же самое можно сказать и обо всех других случаях, где используются не значения слов, а сами слова как ключевые.

Чтобы зарезервировать слово, его надо выделить и щелкнуть на кнопке Зарезервировать слово на вкладке ленты Настройка перевода или воспользоваться аналогичной командой контекстного меню выделенного слова. В открывшемся диалоговом окне Зарезервировать слово можно уточнить написание, указать смысловую категорию, к которой относится данный термин, а также установить флажок Транслитерировать, если нужна транслитерация. Все зарезервированные слова заносятся в список на вкладке Зарезервированные слова на информационной панели, а в самом документе выделяются зеленым цветом.

Если приходится работать с тематически связанными документами или документом, разбитым на несколько отдельных файлов, следует использовать общий список зарезервированных слов. Чтобы сохранить список зарезервированных слов в отдельном файле, следует щелкнуть на вкладке Зарезервированные слова информационной панели правой кнопкой мыши и выбрать в контекстном меню пункт Сохранить список. Для загрузки такого автономного списка в документ используется команда Загрузить список из этого же контекстного меню.

Пополнение словаря. При автоматическом переводе реальных документов часто приходится сталкиваться со словами, которые программа перевода не смогла найти ни в одном из допустимых словарей. Эти слова выделяются в тексте документа красным цветом. Их можно просмотреть на вкладке Незнакомые слова информационной панели. Для того чтобы отобразить эту вкладку, нужно отобразить панель Анализ перевода командой Вид ► Панели ► Анализ перевода.

Слова могут быть неопознаны по разным причинам. В число их могут входить:

- ☐ опечатки в оригинале документа;
- ☐ ошибки распознавания (для документов, преобразованных в электронную форму);

- ☐ собственные имена, требующие резервирования;
- ☐ слова, отсутствующие в словарях.

В первых двух случаях необходимо отредактировать исходный текст, в третьем — зарезервировать слово и только в последнем случае необходимо занести слово в пользовательский словарь. При этом кроме собственно значения слова в переводе необходимо задать грамматические правила изменения форм этого слова и его сочетания с другими словами.

Для того чтобы внести слово в словарь, надо выделить его и щелкнуть на кнопке **Добавить в словарь** на вкладке ленты **Настройка перевода** или воспользоваться контекстным меню слова. В диалоговом окне **Добавить в словарь** нужно выбрать словарь, в который будет внесено это слово, установить режим редактирования (доступны Простой режим и Расширенный режим) и нажать на кнопку **Создать**. После этого, при выборе Простого режима, в окне **Создать словарную статью** надо указать перевод слова, какой частью речи является данное слово, комментарии (рис. 17.6).

Входное слово	Перевод	Часть речи	Комментарий	Активный перевод
▶ РТС		Авто		☒

Рис. 17.6. Средство пополнения пользовательского словаря

Заполните таблицу, описывающую свойства данного слова, и щелкните на кнопке **Сохранить**.

При добавлении слова в словарь с использованием переключателя **Расширенный режим** процесс более длителен, однако он позволяет ввести больше сведений о слове, что в итоге может положительно сказаться на качестве перевода.

Имеющиеся словари можно также просматривать и редактировать. Для этого надо дважды щелкнуть на названии словаря на вкладке **Словари** на информационной панели. Если словарь разрешено редактировать, то он открывается, и на экран выводится список включенных в него слов.

Препроцессоры. Препроцессоры, список которых доступен на вкладке **Препроцессоры** панели свойств программы, позволяют производить предварительную обработку текста для повышения качества перевода и учета особенностей текстов. Например, если в тексте могут встречаться так называемые эмодзи (или смайлики — изображения, состоящие из двоеточий, тире, скобок и других символов, которые используют для передачи настроения в неформальной переписке), к обработке текста можно подключить препроцессор **Smileys**.

Translation Memory. Этот инструмент позволяет сохранять фрагменты текста, типичные для переводимого материала и часто в нем встречающиеся. Такие тексты можно использовать повторно.

Упрощенные средства перевода

Кроме основного приложения, в состав программы PROMT входят дополнительные средства. В частности, это средство Быстрый перевод (рис. 17.7), предназначенное для упрощенного перевода буфера обмена, а также текстов, введенных с клавиатуры в окне любого приложения Windows. Версия Prompt Lite имеет упрощенный интерфейс и ограниченные настройки, однако по возможностям практически полностью соответствует основной версии программы.

Окно программы состоит из двух панелей, верхняя из которых рассматривается как панель оригинала, а нижняя — как панель перевода (см. рис. 17.7).

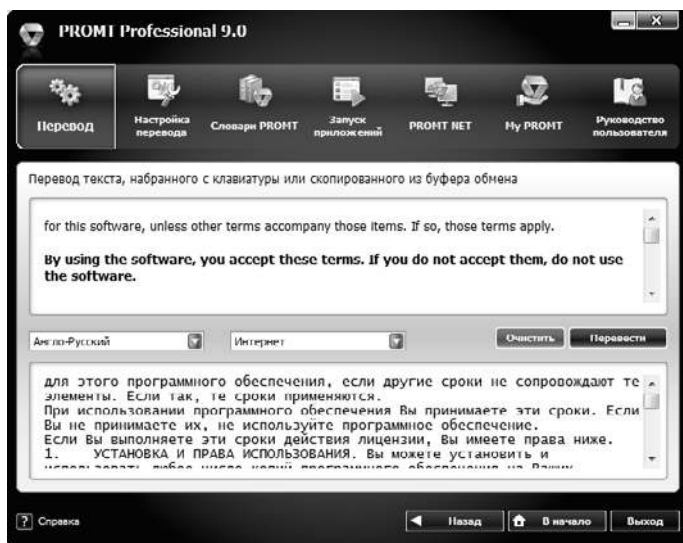


Рис. 17.7. Средство быстрого перевода

Текст для перевода следует ввести в верхнюю панель или вставить из буфера обмена, выбрать направление и основную тематику перевода, после чего нажать на кнопку Перевести. Переведенный текст можно скопировать в буфер обмена.

Интеграция системы перевода с другими приложениями

Для большинства пользователей, нуждающихся в переводе иноязычных текстов лишь от случая к случаю, наибольший интерес вызывает не средство перевода как таковое, а лишь *функции перевода*. Было бы удобно применять такие функции в самых разных программах прямо по ходу дела, не запуская каждый раз мощную внешнюю программу и не перекидывая документ из одной программы в другую и обратно.

При установке PROMT может автоматически интегрироваться с другими приложениями, добавляя в них возможности быстрого перевода без вызова основной программы-переводчика.

В соответствующей программе добавляется новая панель инструментов, вкладка ленты, команда в строке меню, которые можно использовать для выполнения операций по переводу. Текст перевода может заменить исходный, быть добавлен после него или появиться в отдельном окне или в новом документе. В выбранном приложении становятся доступными все основные функции программы *PROMT*.

Наибольшую ценность функция оперативного перевода представляет для документов Интернета. Сегодня на большинстве страниц используется английский язык, поэтому шансы найти нужную информацию именно на англоязычной странице максимальны.

Основные функции по переводу — это перевод выделенного фрагмента страницы (например, гиперссылки) и перевод страницы целиком. По умолчанию перевод текста веб-документа, например, в Internet Explorer открывается в отдельной вкладке.

Еще одна ценная возможность, которую обеспечивают интегрированные функции перевода, — облегчение поиска информации в англоязычных поисковых системах. При таком поиске в Интернете используются ключевые слова, переведенные на английский язык. Чтобы начать поиск, надо щелкнуть в окне браузера правой кнопкой мыши и выбрать в контекстном меню пункт Перевести поисковый запрос.

Диалоговое окно Перевести поисковый запрос (рис. 17.8) содержит три вкладки, обеспечивающие три разных способа формирования запроса на поиск. После того как указаны ключевые слова и выбрана поисковая система, сформированный запрос, содержащий уже переведенные ключевые слова, отображается в специальном поле, чтобы можно было визуально проверить его правильность.

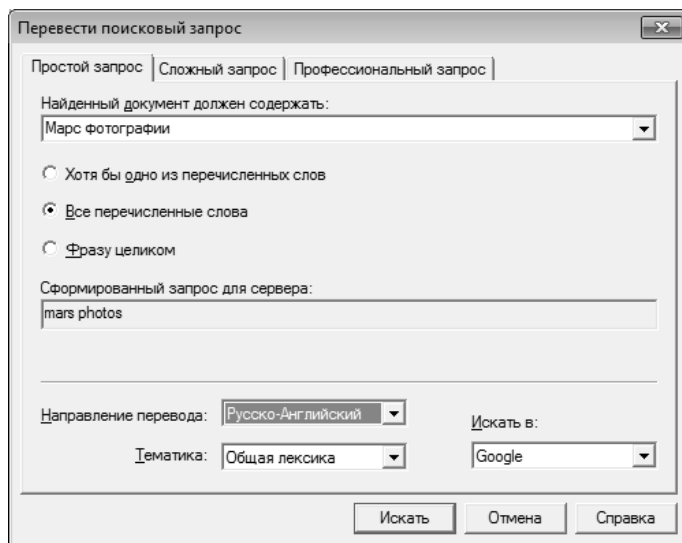


Рис. 17.8. Формирование запроса в Интернет с автоматическим переводом ключевых слов

Практическое занятие

Упражнение 17.5. Автоматический перевод текста

1. Запустите программу PROMT (Пуск ► Все программы ► PROMT ► PROMT Professional 9.0).
2. Нажмите на кнопку Запустить приложение, в появившемся окне нажмите на кнопку Переводчик Promt.
3. Дайте команду Файл ► Открыть. Выберите используемый формат файла в диалоговом окне Конвертировать файл и щелкните на кнопке ОК.
4. Откройте вкладку Словари на информационной панели. Ознакомьтесь со списком установленных словарей.
5. Отключите все словари, кроме одного специализированного тематического словаря по теме документа (и, возможно, пользовательского).
6. Щелкните на кнопке Перевести на вкладке Перевод.
7. Ознакомьтесь с переводом текста. Обратите внимание на вид слов, которые не следовало переводить.
8. Выделите слова, которые не следовало переводить, и щелкните на кнопке За-резервировать слово на вкладке Настройка перевода. Обратите внимание на изменения в отображении оригинала и перевода.
9. Оцените качество автоматического перевода.
10. Сохраните документ во внутреннем формате программы PROMT (Файл ► Сохранить документ). Закройте документ и откройте его снова, убедившись, что в этом случае можно продолжить работу в точности с того места, на котором она была остановлена.
11. Сохраните переведенный текст (Файл ► Сохранить ► Перевод) в формате, учитывающем форматирование документа (Текст в формате RTF). Откройте сохраненный документ с помощью программы WordPad и просмотрите его содержание.
12. Сохраните документ в виде билингвы (Файл ► Сохранить ► Билингву как список абзацев).
13. Откройте сохраненный документ в текстовом редакторе Блокнот. Используя английский и русский варианты текста, попробуйте окончательно сформировать правильный и грамотный перевод исходного документа.

Итог. Мы научились выполнять автоматический перевод документов с помощью программы PROMT. Мы узнали, как влияет выбор словаря на качество перевода. Мы научились исключать из процесса слова, не требующие перевода. Мы познакомились с разными способами сохранения документа и возможностями его дальнейшей обработки.

Упражнение 17.6. Редактирование словаря

1. Запустите программу PROMT (Пуск ► Все программы ► PROMT ► PROMT Professional 9.0).
2. Нажмите на кнопку Запустить приложение, в появившемся окне нажмите на кнопку Переводчик Promt.
3. Дайте команду Файл ► Открыть. Выберите используемый формат файла в диалоговом окне Конвертировать файл и щелкните на кнопке ОК.
4. Щелкните на кнопке Перевести на вкладке Перевод.
5. Выполните команду Вид ► Панели ► Анализ перевода, откройте вкладку Незнакомые слова на информационной панели.
6. Просмотрите список слов, незнакомых программе. Попробуйте разыскать их в англо-русском словаре, чтобы выяснить, какие из них действительно требуют перевода.
7. В списке на вкладке Незнакомые слова на информационной панели дважды щелкните на слове, требующем перевода.
8. В диалоговом окне Добавить в словарь выберите Простой режим и щелкните на кнопке Создать.
9. В диалоговом окне Новая словарная статья укажите часть речи, к которой принадлежит добавляемое слово, и нажмите на кнопку ОК.
10. Состав дальнейших диалоговых окон зависит от выбранной части речи. Заполните поля этих окон, введя необходимые данные.
11. При необходимости внесите в пользовательский словарь и другие слова, требующие перевода, как описано в пп. 8–10.
12. Повторно щелкните на кнопке Перевести на вкладке Перевод. Изучите, как изменился перевод в связи с пополнением словаря.

Итог. Мы увидели, какие действия выполняет программа PROMT при обнаружении в тексте незнакомых ей слов. Мы научились пополнять словари программы, расширяя, таким образом, ее возможности по переводу текстов.

Глава восемнадцатая

Средства автоматизации научно-исследовательских работ

18.1. Компьютер как инструмент научной работы

Вычислительная мощь компьютера позволяет использовать его как средство автоматизации научной работы. Для решения сложных расчетных задач используют программы, написанные специально. В то же время в научной работе встречается широкий спектр задач ограниченной сложности, для решения которых можно использовать универсальные средства.

К такого рода задачам относятся, например, следующие:

- ☐ подготовка научно-технических документов, содержащих текст и формулы, записанные в привычной для специалистов форме;
- ☐ вычисление результатов математических операций, в которых участвуют числовые константы, переменные и размерные физические величины;
- ☐ операции с векторами и матрицами;
- ☐ решение уравнений и систем уравнений (неравенств);
- ☐ статистические расчеты и анализ данных;
- ☐ построение двумерных и трехмерных графиков;
- ☐ тождественные преобразования выражений (в том числе упрощение), аналитическое решение уравнений и систем;
- ☐ дифференцирование и интегрирование, аналитическое и численное;
- ☐ решение дифференциальных уравнений;
- ☐ проведение серий расчетов с разными значениями начальных условий и других параметров.

К универсальным программам, пригодным для решения таких задач, относится, например, программа Mathcad. Это автоматизированная система, позволяющая динамически обрабатывать данные в числовом и аналитическом (формульном) виде. Программа Mathcad сочетает в себе возможности проведения расчетов и подготовки форматированных научных и технических документов.

Научно-технические документы обычно содержат формулы, результаты расчетов в виде таблиц данных или графиков, текстовые комментарии или описания, другие иллюстрации. В программе Mathcad им соответствуют два вида объектов: *формулы* и *текстовые блоки*. Формулы вычисляются с использованием числовых констант, переменных, функций (стандартных и определенных пользователем), а также общепринятых обозначений математических операций. Введенные в документ Mathcad формулы автоматически приводятся к стандартной научно-технической форме записи. Графики, которые автоматически строятся на основе результатов расчетов, также рассматриваются как формулы. Комментарии, описания и иллюстрации размещаются в текстовых блоках, которые игнорируются при проведении расчетов.

Чтобы буквенные обозначения можно было использовать при расчетах по формулам, этим обозначениям должны быть сопоставлены числовые значения. В программе Mathcad буквенные обозначения рассматриваются как переменные, и их значения задаются при помощи оператора присваивания (вводится символом «:=»). Таким же образом можно задать числовые последовательности, аналитически определенные функции, матрицы и векторы.

Если все значения переменных известны, то для вычисления числового значения выражения (скалярного, векторного или матричного) надо подставить все числовые значения и произвести все заданные действия. В программе Mathcad для этого применяют оператор вычисления (вводится символом «=»). В ходе вычисления автоматически используются значения переменных и определения функций, заданные в документе ранее. Удобно задать значения известных параметров, провести вычисления с использованием аналитических формул, результат присвоить некоторой переменной, а затем использовать оператор вычисления для вывода значения этой переменной. Например:

$$\begin{aligned}g &:= 9.8 \\M &:= 3 \\F &:= M \cdot g \\F &= 29.4\end{aligned}$$

Изменение значения любой переменной, коррекция любой формулы означает, что все расчеты, зависящие от этой величины, необходимо проделать заново. Такая необходимость возникает при выборе подходящих значений параметров или условий, поиске оптимального варианта, исследовании зависимости результата от начальных условий. Электронный документ, подготовленный в программе Mathcad, готов к подобной ситуации. При изменении какой-либо формулы программа автоматически производит необходимые вычисления, обновляя изменившиеся значения и графики. Например, если документ содержит формулы $x := 4$; $\sqrt{x} = 2$, то, изменив значение переменной x , мы сразу же увидим, что изменился и результат расчета: $x := 9$; $\sqrt{x} = 3$.

При проведении расчетов с использованием реальных физических величин учитывают их размерность. Чтобы расчет был корректен, все данные должны быть приведены в одну систему единиц — в этом случае результат расчетов получится в этой же системе. Здесь скрывается характерный источник ошибок при расчетах вручную. В программе Mathcad единицы измерения (в любой системе) присоединяют к значению величины с помощью знака умножения. Данные автоматически преобразуются в одну и ту же систему единиц (по умолчанию СИ) и обрабатываются в этом виде. Размерный результат выдается вместе с полученной единицей измерения. Например:

$$\begin{aligned}v &:= 100 \cdot \text{kph} \quad t := 0.5 \cdot \text{yr} \quad (\text{kph} — \text{километры в час, yr} — \text{годы}) \\s &:= v \cdot t \quad s = 4.383 \cdot 10^8 \text{ m} \quad (\text{результат получен в метрах})\end{aligned}$$

При работе с матрицами приходится применять такие операции, как сложение матриц, умножение, транспонирование. Часто возникает необходимость

в обращении матриц и в декомпозиции (разложении в произведение матриц специального вида). Для квадратных матриц представляет интерес поиск собственных значений и собственных векторов. Программа Mathcad позволяет выполнить все эти операции с помощью стандартных обозначений математических операторов (сложение, умножение) или встроенных функций. Например:

$$\begin{aligned} \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}^T &= \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} \\ \left| \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix} \right| &= -1 \\ \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}^{-1} &= \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix} \\ \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}^T \cdot \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix} &= \begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix}. \end{aligned}$$

Уравнения и системы уравнений, возникающие в практических задачах, обычно можно решить только численно. Методы численного решения реализованы и в программе Mathcad. Блок уравнений и неравенств, требующих решения, записывается после ключевого слова *given* (*дано*). При записи уравнений используется знак логического равенства (комбинация клавиш **CTRL+=**). Значения переменных, удовлетворяющие системе уравнений и неравенств, находятся с помощью стандартной функции *find*.

$$\begin{aligned} x &:= 1 \quad y := 0 \\ \text{given} \\ x - y &= 2 \\ \sin(x) &= \sin(y) \\ \text{find}(x, y) &= \begin{pmatrix} 2.571 \\ 0.571 \end{pmatrix}. \end{aligned}$$

При обработке результатов экспериментов часто встречаются задачи статистического анализа серий данных. Для такого рода задач программа Mathcad предоставляет средства интерполяции данных, предсказания дальнейшего поведения функции, а также построения функций заданного вида, наилучшим образом соответствующих имеющемуся набору данных.

При статистическом анализе можно также использовать стандартные функции распределения вероятности и генераторы случайных величин с заданным распределением.

При аналитических вычислениях результат получают в нечисловой форме в результате тождественных преобразований выражений. Простейшие преобразования — это раскрытие скобок, приведение подобных членов, применение тригонометрических тождеств.

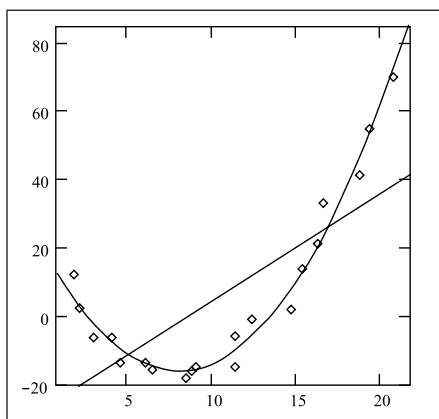


Рис. 18.1. Набор точек аппроксимирован с помощью многочленов первого и второго порядка

Например, выражение $\cos(3 \cdot \text{atan}(x))$ преобразуется в $\frac{4}{(1+x^2)^{\frac{3}{2}}} - \frac{3}{(1+x^2)^{\frac{1}{2}}}$.

Проверку тождественности этого преобразования выполните самостоятельно.

Более сложные преобразования позволяют находить аналитические решения некоторых уравнений и систем. Для такого рода вычислений в программе Mathcad используют оператор аналитического вычисления (клавиатурная комбинация CTRL+.), а также команды меню Symbolics (Аналитические вычисления). Переменные при аналитических вычислениях рассматриваются как неопределенные параметры. Результат можно использовать для анализа решения при различных значениях этих переменных. При аналитическом решении уравнений и систем за одну операцию можно найти все существующие решения. Например:

$$\begin{aligned} &\text{given} \\ &z^3 + 3 \cdot z^2 + 2 \cdot z - 6 = 0 \\ &\text{find}(z) \rightarrow \begin{cases} 1 \\ -2 + i \cdot \sqrt{2} \\ -2 + i \cdot \sqrt{2} \end{cases} \end{aligned}$$

Дифференцирование и интегрирование заданных функций вручную — обычно несложная, но трудоемкая операция. В программе Mathcad для вычисления производной, а также неопределенных и определенных интегралов могут использоваться символические вычисления с помощью меню Symbolics ► Variable (Аналитические вычисления ► Переменная). Если функция не задана аналитически или не позволяет получить первообразную в виде формулы, имеется возможность численного дифференцирования и численного расчета определенных интегралов. Например, при вычислении интеграла

$$\int x \cdot e^{-x} \cdot \cos(x) dx$$

получается правильный результат:

$$-\frac{1}{2}x \exp(-x) \cos(x) + \frac{1}{2}x \exp(-x) \sin(x) + \frac{1}{2} \exp(-x) \sin(x).$$

Численные методы используют и для решения дифференциальных уравнений. С помощью программы Mathcad можно решать уравнения и системы уравнений первого порядка с заданными начальными условиями. Уравнение более высокого порядка надо сначала преобразовать в систему уравнений первого порядка.

18.2. Приемы работы с системой Mathcad

Документ программы Mathcad называется *рабочим листом*. Он содержит объекты: *формулы* и *текстовые блоки*. В ходе расчетов формулы обрабатываются последовательно, слева направо и сверху вниз, а текстовые блоки игнорируются.

Ввод информации осуществляется в месте расположения курсора. Программа Mathcad использует три вида курсоров. Если ни один объект не выбран, используется *крестообразный курсор*, определяющий место создания следующего объекта. При вводе формул используется *уголковый курсор*, указывающий текущий элемент выражения. При вводе данных в текстовый блок применяется *текстовый курсор* в виде вертикальной черты.

Ввод формул

Формулы — основные объекты рабочего листа. Новый объект по умолчанию является формулой. Чтобы начать ввод формулы, надо установить крестообразный курсор в нужное место и начать ввод букв, цифр, знаков операций. При этом создается область формулы, в которой появляется уголковый курсор, охватывающий текущий элемент формулы, например имя переменной (функции) или число. При вводе бинарного оператора по другую сторону знака операции автоматически появляется заполнитель в виде черного прямоугольника. В это место вводят очередной операнд.

Для управления порядком операций используют скобки, которые можно вводить вручную. Уголковый курсор позволяет автоматизировать такие действия. Чтобы выделить элементы формулы, которые в рамках операции должны рассматриваться как единое целое, используют клавишу ПРОБЕЛ. При каждом ее нажатии уголковый курсор «расширяется», охватывая элементы формулы, примыкающие к данному. После ввода знака операции элементы в пределах уголкового курсора автоматически заключаются в скобки.

Элементы формул можно вводить с клавиатуры или с помощью специальных панелей управления. Панели управления (рис. 18.2) открывают с помощью меню View (Вид) или кнопками панели управления Math (Математика). Для ввода элементов формул предназначены следующие панели:

- ❑ панель управления Calculator (Счет) для ввода чисел, знаков типичных математических операций и наиболее часто употребляемых стандартных функций;

- ❑ панель управления Evaluation (Вычисление) для ввода операторов вычисления;
- ❑ панель управления Boolean (Логика) для ввода знаков отношения и логических операций;
- ❑ панель управления Graph (График) для построения графиков;
- ❑ панель управления Matrix (Матрица) для ввода векторов и матриц и задания матричных операций;
- ❑ панель управления Calculus (Исчисление) для задания операций, относящихся к математическому анализу;
- ❑ панель управления Greek (Греческий алфавит) для ввода греческих букв (их можно также вводить с клавиатуры, если сразу после ввода соответствующего латинского символа нажимать сочетание клавиш CTRL+G, например [a][CTRL+G], [W][CTRL+G]);
- ❑ панель управления Symbolic (Аналитические вычисления) для управления аналитическими преобразованиями.



Рис. 18.2. Панели инструментов программы Mathcad для ввода формул

Введенное выражение обычно вычисляют или присваивают переменной. Для вывода результата выражения используют знак вычисления, который выглядит как знак равенства и вводится при помощи кнопки Evaluate Numerically (Вычислить выражение) на панели инструментов Evaluation (Вычисление).

Знак присваивания изображается как «:=», а вводится при помощи кнопки Definition (Определить) на панели инструментов Evaluation (Вычисление). Слева от знака присваивания указывают имя переменной. Оно может содержать латинские и греческие буквы, цифры, символы «'», «_» и «∞», а также описательный индекс. Описательный индекс вводится с помощью символа «.» и изображается как нижний

индекс, но является частью имени переменной, например V_{init} . «Настоящие» индексы, определяющие отдельный элемент вектора или матрицы, задаются по-другому.

Переменную, которой присвоено значение, можно использовать далее в документе в вычисляемых выражениях. Чтобы узнать значение переменной, следует использовать оператор вычисления.

В следующем примере вычислена площадь круга с радиусом 2 (использованы переменные r и s , значение постоянной π определено в программе Mathcad по умолчанию).

$$r := 2 \quad s := \pi \cdot r^2 \quad s = 12.566$$

Ввод текста

Текст, помещенный в рабочий лист, содержит комментарии и описания и предназначен для ознакомления, а не для использования в расчетах. Программа Mathcad определяет назначение текущего блока автоматически при первом нажатии клавиши ПРОБЕЛ. Если введенный текст не может быть интерпретирован как формула, блок преобразуется в текстовый и последующие данные рассматриваются как текст. Создать текстовый блок без использования автоматических средств позволяет команда Insert ► Text Region (Вставка ► Текстовый блок).

Иногда требуется встроить формулу внутрь текстового блока. Для этого служит команда Insert ► Math Region (Вставка ► Формула).

Форматирование формул и текста

Для форматирования формул и текста в программе Mathcad используется панель инструментов Formatting (Форматирование). С ее помощью можно индивидуально отформатировать любую формулу или текстовый блок, задав гарнитуру и размер шрифта, а также полужирное, курсивное или подчеркнутое начертание символов. В текстовых блоках можно также задавать тип выравнивания и применять маркированные и нумерованные списки.

В качестве средств автоматизации используются стили оформления. Выбрать стиль оформления текстового блока или элемента формулы можно из списка Style (Стиль) на панели инструментов Formatting (Форматирование). Для формул и текстовых блоков применяются разные наборы стилей.

Чтобы изменить стиль оформления формулы или создать новый стиль, используется команда Format ► Equation (Формат ► Выражение). Изменение стандартных стилей Variables (Переменные) и Constants (Константы) влияет на отображение формул по всему документу. Стиль оформления имени переменной учитывается при ее определении. Так, переменные x и x рассматриваются как различные и не взаимозаменяемы.

При оформлении текстовых блоков можно использовать более широкий набор стилей. Настройка стилей текстовых блоков производится при помощи команды Format ► Style (Формат ► Стиль).

Работа с матрицами

Векторы и матрицы рассматриваются в программе Mathcad как одномерные и двумерные массивы данных. Число строк и столбцов матрицы задается в диалоговом окне Insert Matrix (Вставка матрицы), которое открывают командой Insert ► Matrix (Вставка ► Матрица). Вектор задается как матрица, имеющая один столбец.

После щелчка на кнопке ОК в формулу вставляется матрица, содержащая вместо элементов заполнители. Вместо каждого заполнителя надо вставить число, переменную или выражение.

Для матриц определены следующие операции: сложение, умножение на число, перемножение и пр. Допустимо использование матриц вместо скалярных выражений: в этом случае предполагается, что указанные действия должны быть применены к каждому элементу матрицы, и результат также представляется в виде матрицы. Например, выражение $M + 3$, где M — матрица, означает, что к каждому элементу матрицы прибавляется число 3. Если требуется явно указать необходимость поэлементного применения операции к матрице, используют знак векторизации, для ввода которого служит кнопка Vectorize (Векторизация) на панели инструментов Matrix (Матрица). Например:

$$\begin{pmatrix} 1 & 3 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 7 & 2 \\ 1 & 2 \end{pmatrix} \quad \text{— обычное произведение матриц};$$

$$\overline{\begin{pmatrix} 1 & 3 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix}} = \begin{pmatrix} 1 & -3 \\ -2 & 1 \end{pmatrix} \quad \text{— поэлементное произведение матриц с использованием векторизации.}$$

Для работы с элементами матрицы используют индексы элементов. Нумерация строк и столбцов матрицы начинается с нуля. Индекс элемента задается числом, переменной или выражением и отображается как нижний индекс. Он вводится после щелчка на кнопке Subscript (Индекс) на панели инструментов Matrix (Матрица).

Пара индексов, определяющих элемент матрицы, разделяется запятой. Иногда (например, при построении графиков) требуется выделить вектор, представляющий собой столбец матрицы. Номер столбца матрицы отображается как верхний индекс, заключенный в угловые скобки, например $M^{<0>}$. Для его ввода используется кнопка Matrix Column (Столбец) на панели инструментов Matrix (Матрица).

Чтобы задать общую формулу элементов матрицы, типа $M_{i,j} := i + j$, используют *диапазоны*. Диапазон фактически представляет собой вектор, содержащий арифметическую прогрессию, определенную первым, вторым и последним элементами. Чтобы задать диапазон, следует указать значение первого элемента, через запятую значение второго и через точку с запятой значение последнего элемента. Точка с запятой при задании диапазона отображается как две точки (..). Диапазон можно использовать как значение переменной, например $x := 0,0.01..1$.

Если разность прогрессии равна единице (то есть элементы просто нумеруются), значение второго элемента и соответствующую запятую опускают. Например, чтобы сформировать по приведенной выше формуле матрицу размером 6×6 , перед этой формулой надо указать $i := 0..5 \quad j := 0..5$. При формировании матрицы путем присвоения значения ее элементам размеры матрицы можно не задавать заранее.

Всем неопределенным элементам автоматически присваиваются нулевые значения. Например, формула $M_{5,5} := 1$ создает матрицу M размером 6×6 , у которой все элементы, кроме расположенного в правом нижнем углу, равны 0.

Стандартные и пользовательские функции

Произвольные зависимости между входными и выходными параметрами задаются при помощи функций. Функции принимают набор параметров и возвращают значение, скалярное или векторное (матричное). В формулах можно использовать стандартные встроенные функции, а также функции, определенные пользователем.

Чтобы использовать функцию в выражении, надо определить значения входных параметров в скобках после имени функции. Имена простейших математических функций можно ввести с панели инструментов Calculator (Счет). Информацию о других функциях можно почерпнуть в справочной системе. Вставить в выражение стандартную функцию можно при помощи команды Insert ► Function (Вставка ► Функция). В диалоговом окне Insert Function (Вставка функции) слева выбирается категория, к которой относится функция, а справа — конкретная функция. В нижней части окна выдается информация о выбранной функции. При вводе функции через это диалоговое окно автоматически добавляются скобки и заполнители для значений параметров.

Пользовательские функции должны быть сначала определены. Определение задается при помощи оператора присваивания. В левой части указывается имя пользовательской функции и, в скобках, формальные параметры — переменные, от которых она зависит. Справа от знака присваивания эти переменные должны использоваться в выражении. При использовании пользовательской функции в последующих формулах ее имя вводят вручную. В диалоговом окне Insert Function (Вставка функции) оно не отображается.

Решение уравнений и систем

Для численного поиска корней уравнения в программе Mathcad используется функция *root*. Она служит для решения уравнений вида $f(x) = 0$, где $f(x)$ — выражение, корни которого нужно найти, а x — неизвестное. Для поиска корней с помощью функции *root* надо присвоить искомой переменной начальное значение, а затем вычислить корень при помощи вызова функции:

$$\text{root}(f(x), x)$$

Здесь $f(x)$ — функция переменной x , используемой в качестве второго параметра. Функция *root* возвращает значение независимой переменной, обращающее функцию $f(x)$ в 0. Например:

$$x := 1$$

$$\text{root}(2 \cdot \sin(x) - x, x) = 1.895$$

Если уравнение имеет несколько корней (как в данном примере), то результат, выдаваемый функцией *root*, зависит от выбранного начального приближения.

Если надо решить систему уравнений (неравенств), используют так называемый *блок решения*, который начинается с ключевого слова *given* (*дано*) и заканчивается вызовом функции *find* (*найти*). Между ними располагают «логические утверждения», задающие ограничения на значения искомых величин, иными словами, уравнения и неравенства. Всем переменным, используемым для обозначения неизвестных величин, должны быть заранее присвоены начальные значения.

Чтобы записать уравнение, в котором утверждается, что левая и правая части равны, используется знак логического равенства — кнопка **Equal to** (**Равно**) на панели инструментов **Boolean** (**Логика**). Другие знаки логических условий также можно найти на этой панели.

Заканчивается блок решения вызовом функции *find*, у которой в качестве аргументов должны быть перечислены искомые величины. Эта функция возвращает вектор, содержащий вычисленные значения неизвестных. Например:

$$\begin{aligned} &x := 0 \quad y := 0 \\ &\text{given} \\ &x + y = 1 \\ &x^2 + y^2 = 4 \\ &\text{find}(x, y) = \begin{pmatrix} 1,823 \\ 0,823 \end{pmatrix}. \end{aligned}$$

Построение графиков

Чтобы построить двумерный график в координатных осях X – Y , надо дать команду **Insert** ► **Graph** ► **X-Y Plot** (**Вставка** ► **График** ► **Декартовы координаты**). В области размещения графика находятся заполнители для указания отображаемых выражений и диапазона изменения величин. Заполнитель у середины оси координат предназначен для переменной или выражения, отображаемого по этой оси. Обычно используют диапазон или вектор значений. Граничные значения по осям выбираются автоматически в соответствии с диапазоном изменения величины, но их можно задать и вручную.

В одной графической области можно построить несколько графиков. Для этого надо у соответствующей оси перечислить несколько выражений через запятую.

Разные кривые изображаются разным цветом, а для форматирования графика надо дважды щелкнуть на области графика. Для управления отображением построенных линий служит вкладка **Traces** (**Линии**) в открывшемся диалоговом окне. Текущий формат каждой линии приведен в списке, а под списком расположены элементы управления, позволяющие изменять формат. Поле **Legend Label** (**Описание**) задает описание линии, которое отображается только при сбросе флажка **Hide Legend** (**Скрыть описание**). Список **Symbol** (**Символ**) позволяет выбрать маркеры для отдельных точек, список **Line** (**Тип линии**) задает тип линии, список **Color** (**Цвет**) — цвет. Список **Type** (**Тип**) определяет способ связи отдельных точек, а список **Weight** (**Толщина**) — толщину линии.

Точно так же можно построить и отформатировать график в полярных координатах. Для его построения надо дать команду Insert ► Graph ► Polar Plot (Вставка ► График ► Полярные координаты).

Для построения простейшего трехмерного графика необходимо задать матрицу значений. Отобразить эту матрицу можно в виде поверхности — Insert ► Graph ► Surface Plot (Вставка ► График ► Поверхность), столбчатой диаграммы — Insert ► Graph ► 3D Bar Plot (Вставка ► График ► Столбчатая диаграмма) или линий уровня — Insert ► Graph ► Contour Plot (Вставка ► График ► Линии уровня).

Для отображения векторного поля при помощи команды Insert ► Graph ► Vector Field Plot (Вставка ► График ► Поле векторов) значения матрицы должны быть комплексными. В этом случае в каждой точке графика отображается вектор с координатами, равными действительной и мнимой частям элемента матрицы. Во всех этих случаях после создания области графика необходимо указать вместо заполнителя имя матрицы, содержащей необходимые значения.

Для построения параметрического точечного графика командой Insert ► Graph ► 3D Scatter Plot (Вставка ► График ► Точки в пространстве) необходимо задать три вектора с одинаковым числом элементов, которые соответствуют x -, y - и z -координатам точек, отображаемых на графике. В области графика эти три вектора указываются внутри скобок через запятую.

Аналогичным образом можно построить поверхность, заданную параметрически. Для этого надо задать три матрицы, содержащие соответственно x -, y - и z -координаты точек поверхности. Теперь надо дать команду построения поверхности Insert ► Graph ► Surface Plot (Вставка ► График ► Поверхность) и указать в области графика эти три матрицы в скобках и через запятую. Таким образом можно построить практически любую криволинейную поверхность (например, представленную на рис. 18.3), в том числе с самопересечениями.

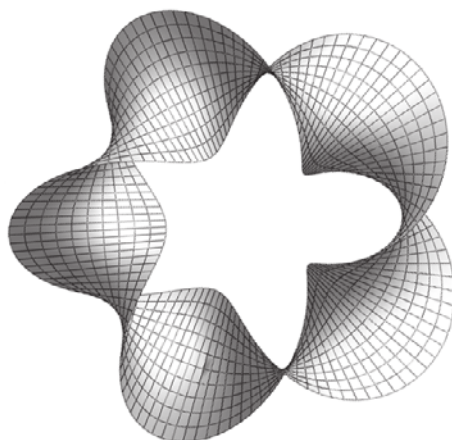


Рис. 18.3. Пятикратно перекрученная замкнутая лента, заданная параметрически

Диалоговое окно для форматирования трехмерных графиков также открывают двойным щелчком на области графика.

Аналитические вычисления

С помощью аналитических вычислений находят аналитические или полные решения уравнений и систем, а также проводят преобразования сложных выражений (например, упрощение). Иначе говоря, при таком подходе можно получить нечисловой результат. В программе Mathcad конкретные значения, присвоенные переменным, при этом игнорируются — переменные рассматриваются как неопределенные параметры. Команды для выполнения аналитических вычислений в основном сосредоточены в меню **Symbolics** (Аналитические вычисления).

Чтобы упростить выражение (или часть выражения), надо выбрать его при помощи уголкового курсора и дать команду **Symbolics** ▶ **Simplify** (Аналитические вычисления ▶ Упростить). При этом выполняются арифметические действия, сокращаются общие множители и приводятся подобные члены, применяются тригонометрические тождества, упрощаются выражения с радикалами, а также выражения, содержащие прямую и обратную функции (типа $e^{\ln x}$). Некоторые действия по раскрытию скобок и упрощению сложных тригонометрических выражений требуют применения команды **Symbolics** ▶ **Expand** (Аналитические вычисления ▶ Раскрыть).

Команду **Symbolics** ▶ **Simplify** (Аналитические вычисления ▶ Упростить) применяют и в более сложных случаях. Например, с ее помощью можно:

- вычислить предел числовой последовательности, заданной общим членом;
- найти общую формулу для суммы членов числовой последовательности, заданной общим членом;
- вычислить производную данной функции;
- найти первообразную данной функции или значение определенного интеграла.

Другие возможности меню **Symbolics** (Аналитические вычисления) состоят в выполнении аналитических операций, ориентированных на переменную, использованную в выражении. Для этого надо выделить в выражении переменную и выбрать команду из меню **Symbolics** ▶ **Variable** (Аналитические вычисления ▶ Переменная). Команда **Solve** (Решить) ищет корни функции, заданной данным выражением, например, если выделить уголковым курсором переменную x в выражении $a \cdot x^2 + b \cdot x + c$, то в результате применения команды **Symbolics** ▶ **Variable** ▶ **Solve** (Аналитические вычисления ▶ Переменная ▶ Решить) будут найдены все корни:

$$\left[\frac{1}{(2 \cdot a)} \cdot \left[-b + \sqrt{(b^2 - 4 \cdot a \cdot c)} \right] \right] \\ \left[\frac{1}{(2 \cdot a)} \cdot \left[-b - \sqrt{(b^2 - 4 \cdot a \cdot c)} \right] \right].$$

Другие возможности использования этого меню включают:

- аналитическое дифференцирование и интегрирование: **Symbolics** ▶ **Variable** ▶ **Differentiate** (Аналитические вычисления ▶ Переменная ▶ Дифференцировать) и **Symbolics** ▶ **Variable** ▶ **Integrate** (Аналитические вычисления ▶ Переменная ▶ Интегрировать);

- ❑ замена переменной: Symbolics ▶ Variable ▶ Substitute (Аналитические вычисления ▶ Переменная ▶ Подставить) — вместо переменной подставляется содержимое буфера обмена;
- ❑ разложение в ряд Тейлора: Symbolics ▶ Variable ▶ Expand to Series (Аналитические вычисления ▶ Переменная ▶ Разложить в ряд);
- ❑ представление дробно-рациональной функции в виде суммы простых дробей с линейными и квадратичными знаменателями: Symbolics ▶ Variable ▶ Convert to Partial Fraction (Аналитические вычисления ▶ Переменная ▶ Преобразовать в простые дроби).

Наконец, самым мощным инструментом аналитических вычислений является оператор аналитического вычисления, который вводится с помощью кнопки Evaluate Symbolically (Вычислить аналитически) на панели инструментов Evaluation (Вычисление). Его можно, например, использовать для аналитического решения системы уравнений и неравенств. Блок решения задается точно так же, как при численном решении (хотя начальные значения переменных можно не задавать), а последняя формула блока должна выглядеть как $\text{find}(x, y, \dots) \rightarrow$, где в скобках приведен список искомых величин, а далее следует знак аналитического вычисления, отображаемый в виде стрелки, направленной вправо.

Любое аналитическое вычисление можно применить с помощью ключевого слова. Для этого используют кнопку Symbolic Keyword Evaluation (Вычисление с ключевым словом) на панели инструментов Evaluation (Вычисление). Ключевые слова вводятся через панель инструментов Symbolics (Аналитические вычисления). Они полностью охватывают возможности, заключенные в меню Symbolics (Аналитические вычисления), позволяя также задавать дополнительные параметры.

Практическое занятие

Упражнение 18.1. Простые вычисления с использованием программы Mathcad

Задача. Найти ребро куба, равновеликого шару, площадь поверхности которого равна площади боковой поверхности прямого кругового конуса, у которого высота вдвое меньше, чем длина образующей. Объем этого конуса равен 1.

Анализ. Основные геометрические формулы, используемые при расчете.

$$\text{Объем конуса} - V = \frac{1}{3} \pi r^2 h.$$

$$\text{Площадь боковой поверхности конуса} - S = \pi r l.$$

Соотношение в конусе между радиусом основания, высотой и длиной образующей — $r^2 + h^2 = l^2$.

$$\text{Площадь поверхности шара} - V = 4\pi R^2.$$

$$\text{Объем шара} - V = \frac{4}{3} \pi r^3. \quad \text{Объем куба} - V = a^3.$$

1. Запустите программу Mathcad через Главное меню (Пуск ▶ Все программы ▶ Mathcad 15).

- Откройте панель инструментов Calculator (Счет) щелчком на кнопке Calculator Toolbar (Панель инструментов Счет) на панели инструментов Math (Математика) или с помощью команды View ► Toolbars ► Calculator (Вид ► Панели инструментов ► Счет).
- Для удобства расчета будем обозначать каждую из вычисляемых величин отдельной переменной. Объем конуса обозначим как V и присвоим ему значение 1. Оператор присваивания вводится символом «:=» или кнопкой Definition (Определить) на панели инструментов Calculator (Счет). Итак, надо ввести $V:=1$. В документе появится полноценный оператор присваивания.

$$V:=1$$

- Путем несложных преобразований получим, что радиус основания конуса можно вычислить по формуле

$$r = \sqrt[3]{\frac{V \cdot \sqrt{3}}{\pi}}.$$

Вводить эту формулу следует слева направо. Порядок ввода этой формулы следующий. Сначала введите знак корня произвольной степени: кнопка Nth Root (Корень данной степени) на панели инструментов Calculator (Счет) или комбинация клавиш CTRL+ $\sqrt{}$. Щелкните на черном квадратике, стоящем на месте показателя степени, и введите цифру 3. Щелкните на квадратике, замещающем подкоренное выражение, нажмите клавиши V*. Введите знак квадратного корня: кнопка Square Root (Квадратный корень) на панели инструментов Calculator (Счет) или клавиша $\sqrt{}$ — и цифру 3. Прежде чем вводить знаменатель, дважды нажмите клавишу ПРОБЕЛ. Обратите внимание на синий уголок, который указывает на текущее выражение. Предполагается, что знак операции связывает выбранное выражение со следующим. В данном случае это безразлично, но в целом этот прием позволяет вводить сложные формулы, избегая ручного ввода дополнительных скобок. Нажмите клавишу /. Чтобы ввести число π , можно воспользоваться комбинацией клавиш CTRL+SHIFT+P или соответствующей кнопкой на панели инструментов Calculator (Счет).

На экране появится следующая надпись: $r = \sqrt[3]{\frac{V \cdot \sqrt{3}}{\pi}}$.

- Введите формулы для вычисления длины образующей и площади боковой поверхности конуса:

$$l = \frac{r \cdot 2}{\sqrt{3}}; \quad S = \pi \cdot r \cdot l.$$

Указание знака умножения между переменными обязательно, так как иначе программа Mathcad сочтет, что указана одна переменная с именем из нескольких букв.

- Для вычисления радиуса шара R введите формулу $R = \sqrt{\frac{S}{4\pi}}$.

7. Для вычисления объема шара введите формулу $W = \frac{4}{3}\pi R^2$.

Использовать переменную V во второй раз не следует, так как теперь мы определяем совершенно другой объем.

8. Заключительная формула $a = \sqrt[3]{W}$ позволит получить окончательный результат. После этого снова наберите имя переменной a и нажмите клавишу = или щелкните на кнопке Evaluate Expression (Вычислить выражение) на панели инструментов Calculator (Счет). После формулы появится знак равенства и вычисленный результат.

$$a = 0.7102.$$

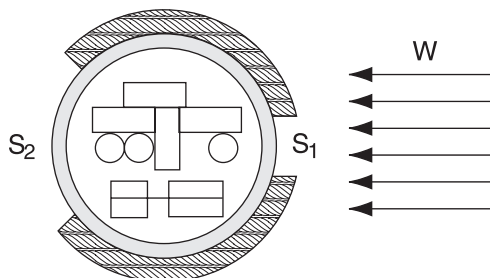
Вычислять можно как действительные, так и комплексные выражения. Обозначение мнимой единицы (i) следует вводить непосредственно после числового коэффициента, который нельзя опускать, даже если он равен единице.

9. Вернитесь к самому первому выражению и отредактируйте его. Вместо значения 1 присвойте переменной значение 8. Сразу же перейдите к последней введенной формуле и обратите внимание, что результат расчета сразу же стал отражать новые начальные данные.

Итог. Мы познакомились с методикой простейших вычислений в программе MathCad. Описанная техника позволяет использовать эту программу как «интеллектуальный калькулятор» для автоматического расчета по известным формулам. Особенностью программы Mathcad является возможность практически мгновенного перерасчета с другими начальными данными.

Упражнение 18.2. Физические вычисления с использованием единиц измерения

Постановка задачи. Теплоизолированный космический аппарат, находящийся на орбите Земли, имеет на борту приборы с электрической мощностью, которая может изменяться в ходе работы от $N_1 = 75$ Вт (дежурный режим) до $N_2 = 200$ Вт (сеанс связи). С целью обеспечения предсказуемого теплового режима в теплоизоляции сделано отверстие площадью S_1 , на которое попадает поток солнечной энергии $W = 1400$ Вт/м². Полученная энергия излучается аппаратом через это и дополнительное отверстие в теплоизоляции с площадью S_2 в режиме «черного тела». Каковы должны быть площади отверстий, если допустимый диапазон температур для оборудования, расположенного в аппарате, составляет 20–30 °С?



Анализ задачи. Минимальная температура аппаратуры соответствует режиму минимального тепловыделения. В этом случае поступающая мощность $Q_1 = WS_1 + N_1$. Излучаемая мощность $Q_2 = \sigma T_1^4 (S_1 + S_2)$, где T_1 — минимальная допустимая температура в градусах кельвина. В условиях теплового баланса эти мощности должны быть равны.

Режим максимального тепловыделения соответствует максимальной температуре аппаратуры. В этом случае $WS_1 + N_2 = \sigma T_2^4 (S_1 + S_2)$.

Используя два полученных уравнения, получаем:

$$S_1 = \frac{(N_2 T_1^4 - N_1 T_2^4)}{W \cdot (T_2^4 - T_1^4)}; \quad S_2 = \frac{W \cdot (N_2 - N_1) - \sigma (N_2 T_1^4 - N_1 T_2^4)}{\sigma W \cdot (T_2^4 - T_1^4)}$$

1. Запустите программу Mathcad.
2. Введите значения известных величин, присвоив их переменным с соответствующими именами. Вместо нижних индексов используйте просто дополнительную цифру в названии переменной.

$$W := 1400 \frac{\text{watt}}{\text{m}^2}$$

$$N_1 := 75 \cdot \text{watt} \quad N_2 := 200 \cdot \text{watt}$$

$$T_1 := (20 + 273) \cdot \text{K} \quad T_2 := (30 + 273) \cdot \text{K}$$

3. Обозначения физических единиц присоединяйте к соответствующим значениям, вводя их с клавиатуры вслед за значением. Если нужное обозначение неизвестно, используйте команду Insert ► Unit (Вставка ► Единица измерения). Измеряемая величина выбирается в списке Dimension (Размерность), а нужная единица измерения — в списке Unit (Единица измерения).

Присвойте переменной s значение постоянной Стефана—Больцмана $\left(5,67 \cdot 10^{-8} \frac{\text{Вт}}{\text{м}^2 \cdot \text{К}^4}\right)$.

Чтобы ввести греческую букву, используйте панель инструментов Greek (Греческий алфавит) или введите соответствующую латинскую букву (в данном случае «s») и сразу же нажмите комбинацию клавиш CTRL+G. Так как специальной единицы для размерности этой константы не существует, ее следует составить из стандартных единиц с помощью умножения и деления.

4. Введите полученные в ходе анализа формулы для вычисления площадей отверстий, присвоив полученные значения переменным S_1 и S_2 .

$$S_1 := \frac{(N_2 T_1^4 - N_1 T_2^4)}{W \cdot (T_2^4 - T_1^4)}; \quad S_2 := \frac{W \cdot (N_2 - N_1) - s \cdot (N_2 T_1^4 - N_1 T_2^4)}{s W \cdot (T_2^4 - T_1^4)}$$

Изменение значений параметров, заданных в условии задачи, приводит к автоматическому перерасчету формул. В частности, исследуйте, изменяя значение переменной W , как изменяются требования к такому методу терморегуляции

при удалении аппарата от Солнца и приближении к нему ($W = 2700 \frac{\text{Вт}}{\text{м}^2}$; на орбите Марса $W = 500 \frac{\text{Вт}}{\text{м}^2}$).

5. Обратите внимание, что результат содержит единицы измерения в соответствии с системой единиц СИ. Используемая система единиц отображается в диалоговом окне Insert Unit (Вставка единиц измерения).
6. Чтобы изменить используемую систему единиц, дайте команду Math ► Options (Математика ► Параметры) и в открывшемся диалоговом окне Math Options (Параметры расчета) выберите вкладку Unit System (Система единиц). Выберите систему CGS и посмотрите, как изменились результаты (они теперь выражаются в квадратных сантиметрах). Если, например, выбрать американскую систему единиц (U.S.), то результат будет выражен в квадратных футах.

Итог. Мы научились производить вычисления с использованием реальных размерных физических величин, а также производить преобразование данных из одной системы единиц в другую. Это позволяет немедленно получать результат в наиболее удобной форме.

Упражнение 18.3. Векторы и матрицы

Задача. Разложить вектор $\vec{V} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ по нормированным собственным векторам матрицы $M = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 6 & 1 \\ 2 & 1 & 1 \end{bmatrix}$.

Анализ. Первый этап решения задачи состоит в нахождении собственных значений и собственных векторов данной матрицы. Затем необходимо найти вектор $\vec{T}$, такой, что $S \times \vec{T} = \vec{V}$, где S — матрица, столбцы которой представляют собой собственные векторы матрицы M .

1. Запустите программу Mathcad.
2. Создайте матрицу M . Начните запись оператора присваивания, а для ввода правой части нажмите комбинацию клавиш CTRL+M, воспользуйтесь командой Insert ► Matrix (Вставка ► Матрица) или щелкните на кнопке Matrix or Vector (Матрица или вектор) на панели инструментов Matrix (Матрица).
3. В открывшемся диалоговом окне Insert Matrix (Вставка матрицы) укажите число строк и столбцов (по три) и щелкните на кнопке OK.
4. Введите значения элементов матрицы в отведенные места.

Аналогичным образом сформируйте вектор $\vec{V}$. Он будет представлять собой матрицу, имеющую только один столбец.

5. Собственные значения квадратной матрицы можно получить при помощи функции *eigenvals*. Результатом ее работы является вектор собственных значений, присвойте его переменной L .
6. Функция *eigenvec* позволяет получить собственный вектор, соответствующий данному собственному значению. Ей нужны два параметра: матрица, для которой ищется собственный вектор, и собственное значение, которому он соответствует. Чтобы записать собственные векторы в качестве столбцов матрицы S , надо присвоить вычисленное значение столбцу матрицы. Столбцы матрицы в программе Mathcad выбираются специальным верхним индексом, заключенным в угловые скобки. Чтобы ввести номер столбца, нажмите комбинацию клавиш **CTRL+6** или щелкните на кнопке **Matrix Column (Столбец)** на панели инструментов **Matrix (Матрица)**, после чего введите номер нужного столбца матрицы. Будьте внимательны — столбцы и строки матрицы нумеруются начиная с нуля.

$$S^{<0>}$$

7. В правой части оператора присваивания надо указать собственное значение матрицы. Собственные значения являются элементами вектора L . Номер элемента указывается как нижний индекс. Для ввода нижнего индекса нажмите клавишу **[** или воспользуйтесь кнопкой **Subscript (Индекс)** на панели инструментов **Matrix (Матрица)**. Итоговый оператор для первого собственного вектора будет выглядеть следующим образом:

$$S^{<0>} := \text{eigenvec}(M, L_0).$$

Аналогично задайте операторы для второго и третьего собственных значений.

8. Для нахождения коэффициентов при собственных векторах в разложении необходимо решить систему линейных уравнений. Ее удобно записать в матричной форме. Создайте вектор T с тремя элементами. Величины этих элементов значения не имеют.
9. Запишите ключевое слово *given*.
10. Ниже запишите матричное уравнение $S \cdot T = V$. Знак логического равенства введите с помощью комбинации клавиш **CTRL+=**.
11. Найдите коэффициенты в разложении при помощи функции *find*.

$$\text{find}(T) = \begin{bmatrix} 0,836 \\ 0,334 \\ 0,148 \end{bmatrix}.$$

Итог. Мы научились производить операции с векторами и матрицами, использовать соответствующие функции, выделять столбцы матриц и отдельные элементы. Матричная запись часто позволяет представить задачу в более удобной форме.

Упражнение 18.4. Аналитические вычисления

Задача 1. На приведенной схеме сопротивление RR является переменным. Определить, как меняется ток между точками A и B в зависимости от величины этого сопротивления.

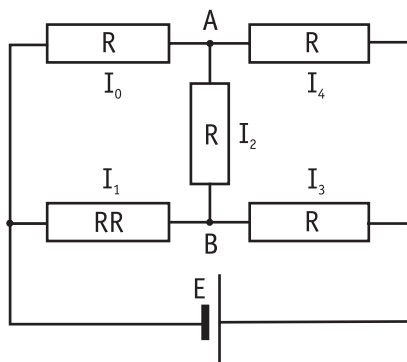


Рис. 18.4. Исследуемая электрическая схема

Анализ. Перенумеровав сопротивления в указанном порядке и воспользовавшись законами Кирхгофа, получим систему уравнений, позволяющую найти величины токов.

$$\begin{cases} I_0 + I_2 = I_4 \\ I_1 = I_2 = I_3 \\ RR \cdot I_1 + R \cdot I_2 - R \cdot I_0 = E \\ R \cdot I_2 + R \cdot I_4 - R \cdot I_3 = 0 \\ R \cdot I_0 + R \cdot I_4 = E \end{cases}$$

Эту систему надо решить, не подставляя конкретных значений вместо параметров R , RR и E .

1. Запустите программу Mathcad.
2. Введите ключевое слово *given*.
3. Введите уравнения системы, полученной в ходе анализа. Обозначьте неизвестные токи переменными I_0 , I_1 , I_2 , I_3 , I_4 . Фиксированное сопротивление R обозначьте переменной $R0$. Обратите внимание, что присваивать начальные значения токов или задавать значения переменных $R0$, RR и E не требуется.
4. Введите функцию *find*, перечислив в качестве параметров неизвестные I_0 , I_1 , I_2 , I_3 , I_4 . Затем введите оператор аналитического вычисления, который выглядит как стрелка, направленная вправо, и вводится комбинацией клавиш CTRL+. или кнопкой Evaluate Symbolically (Вычислить аналитически) на панели инструментов Evaluation (Вычисление).

5. Щелкните за пределами данного блока, и программа Mathcad произведет аналитическое решение системы уравнений.

Полученный результат позволяет провести полный анализ схемы.

$$\text{find}(I_0, I_1, I_2, I_3, I_4) \rightarrow \begin{pmatrix} E \cdot \frac{(3 + RR + R0)}{(R0) \cdot (5 \cdot RR + 3 \cdot R0)} \\ 4 \cdot \frac{E}{(5 \cdot RR + 3 \cdot R0)} \\ E \cdot \frac{(-RR + R0)}{(R0 \cdot (5 \cdot RR + 3 \cdot R0))} \\ E \cdot \frac{(RR + 3 \cdot R0)}{(R0 \cdot (5 \cdot RR + 3 \cdot R0))} \\ 2 \cdot R \cdot \frac{(-RR + R0)}{(R0 \cdot (5 \cdot RR + 3 \cdot R0))} \end{pmatrix}.$$

Задача 2. Найти все корни уравнения:

$$(1 + y - y^2)^2 + y = 2.$$

Анализ. Это уравнение четвертого порядка. Легко подобрать один корень ($y = 1$). Остающееся уравнение третьего порядка не имеет рациональных корней, так что поиск других корней этого уравнения — дело непростое. Неясно даже, сколько еще действительных корней имеет данное уравнение. Результаты численного решения зависят от подбора начального приближения и поэтому не гарантируют отыскания всех корней уравнения. Мы же решим это уравнение аналитически.

1. Введите заданное уравнение. Чтобы раскрыть скобки, дайте команду **Symbolics** ▶ **Simplify** (Аналитические вычисления ▶ Упростить).
2. Выделите в полученном уравнении независимую переменную (в данном случае y) и дайте команду **Symbolics** ▶ **Variable** ▶ **Solve** (Аналитические вычисления ▶ Переменная ▶ Решить).

Программа Mathcad выдаст вектор, элементами которого являются корни данного уравнения.

3. Полученный результат содержит сложные комплексные выражения и его невозможно применить с пользой (все еще непонятно, являются ли корни действительными или комплексными). Чтобы разделить действительную и мнимую части, выделите результат вычисления целиком и дайте команду **Symbolics** ▶ **Evaluate** ▶ **Complex** (Аналитические вычисления ▶ Вычислить ▶ В комплексном виде). Если программа Mathcad не справится с преобразованием всего набора корней целиком, выполните преобразование корней поочередно: дайте команду **Symbolics** ▶ **Evaluate** ▶ **Complex** (Аналитические вычисления ▶ Вычислить ▶ В комплексном виде), поочередно выделив каждый из корней, записанных в комплексном виде.

4. Теперь полученное выражение надо упростить. Выделив его целиком, дайте команду **Symbolics ▶ Simplify** (Аналитические вычисления ▶ Упростить). Выражение станет существенно проще, причем станет понятно, что все корни уравнения действительные (все мнимые компоненты сократятся).
5. Последний шаг — раскрытие скобок, в данном случае упрощение аргументов тригонометрических функций. Для этого примените команду **Symbolics ▶ Expand** (Аналитические вычисления ▶ Раскрыть). Полученная запись — наилучшее представление точного решения, которое можно получить с помощью программы Mathcad.
6. Чтобы получить результат в числовом виде, достаточно ввести в конце выражения (итогового или на любой из предыдущих стадий) команду вычисления (=).

Мы научились использовать программу Mathcad для выполнения аналитических вычислений. Это позволяет получать точные решения задач, содержащих переменные параметры, анализировать полученные результаты, а также получать полный набор решений для некоторых типов уравнений.

$$\begin{bmatrix} 1 \\ \frac{1}{3} \cdot 7^{\frac{1}{2}} \cdot \cos\left(\frac{1}{3} \cdot a \tan(3 \cdot 3^{\frac{1}{2}})\right) + \frac{1}{3} \cdot 7^{\frac{1}{2}} \cdot \sin\left(\frac{1}{3} \cdot a \tan(3 \cdot 3^{\frac{1}{2}})\right) \cdot 3^{\frac{1}{2}} + \frac{1}{3} \\ -\frac{2}{3} \cdot 7^{\frac{1}{2}} \cdot \cos\left(\frac{1}{3} \cdot a \tan(3 \cdot 3^{\frac{1}{2}})\right) + \frac{1}{3} \\ \frac{1}{3} \cdot 7^{\frac{1}{2}} \cdot \cos\left(\frac{1}{3} \cdot a \tan(3 \cdot 3^{\frac{1}{2}})\right) - \frac{1}{3} \cdot 7^{\frac{1}{2}} \cdot \sin\left(\frac{1}{3} \cdot a \tan(3 \cdot 3^{\frac{1}{2}})\right) \cdot 3^{\frac{1}{2}} + \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 1 \\ 1,802 \\ -1,247 \\ 0,445 \end{bmatrix}.$$

Упражнение 18.5. Анализ результатов испытаний

Задача. К пружине последовательно подвешивали грузы массой 1, 2, 3, ..., 20 кг. В результате был получен список величин удлинения пружины (в миллиметрах). Определить основные статистические параметры полученного набора измерений. Рассчитать жесткость пружины и массу узла, использованного для крепления грузов к пружине, воспользовавшись методом наименьших квадратов.

Таблица измерений:

Вес, кг	1	2	3	4	5	6	7	8	9	10
Растяжение, мм	3,5	6,9	9,2	12,2	13,5	173	22,1	242	27,3	29,6
Вес, кг	11	12	13	14	15	16	17	18	19	20
Растяжение, мм	31,8	37,6	39,6	42,8	45,5	46,6	52,1	52,5	56,7	62,4

Анализ. Для решения этой задачи достаточно использовать стандартные средства статистических вычислений, имеющиеся в программе Mathcad. Теоретически растяжение пружины определяется формулой $k > x = (m + m_0) > g$. Если определить статистическими методами коэффициенты a и b в уравнении $x = a > m + b$, то получим:

$$k = \frac{g}{a}, \quad m_0 = \frac{b}{a}.$$

1. Запустите программу Mathcad.
2. Введите таблицу данных, предназначенных для статистического анализа, как матрицу с двумя столбцами, первый из которых содержит веса грузов, а второй — значения растяжения пружины.
3. Определите число точек в наборах данных с помощью функции *rows*.

$$n := \text{rows}(\text{data}) \quad n = 20$$

4. Вычислите среднее растяжение пружины в ходе эксперимента с помощью функции *mean*.

$$Y := \text{data}^{<1>} \quad \text{mean}(Y) = 31.695$$

5. Вычислите медиану значений растяжения пружины при помощи функции *median*.

$$\text{median}(Y) = 30.7$$

6. Вычислите среднеквадратичное отклонение и дисперсию величины растяжения пружины при помощи функции *stdev*.

$$\text{stdev}(Y) = 17.39625 \quad \text{stdev}(Y)^2 = 302.62947$$

7. Определите коэффициенты линейного уравнения являющегося наилучшим приближением для данных наборов данных. Функция *slope* позволяет вычислить коэффициент наклона прямой, а функция *intercept* — свободный член.

$$X := \text{data}^{<0>}$$

$$b_0 := \text{intercept}(X, Y) \quad b_0 = 0.07421$$

$$b_1 := \text{slope}(X, Y) \quad b_1 = 3.0115$$

8. Определите жесткость пружины.

$$k = 3,254 > 10^3 \text{ (Н/м)}$$

9. Определите массу узла крепления.

$$m = 24,64 \text{ (г)}$$

10. Сохраните созданный документ для использования в следующем упражнении.

Итог. Мы научились применять функции, используемые для статистического анализа данных. Программа Mathcad содержит и другие функции аналогичного назначения, которые можно использовать для интерполяции и экстраполяции данных, а также их сглаживания.

Упражнение 18.6. Построение графиков

Задача. Используя результаты, полученные в предыдущем упражнении, построить график, отображающий экспериментальные данные и аппроксимирующую зависимость. Построить другой график, отображающий величину отклонения экспериментальных значений от аппроксимирующей прямой.

Анализ. Для построения графика можно использовать функцию, заданную набором данных или формулой. Формулы для функций, полученных в результате проделанных расчетов, необходимо определить, прежде чем их можно будет использовать при построении графика.

1. Запустите программу Mathcad.
2. Загрузите документ, созданный в предыдущем упражнении.
3. Переместите точку ввода в нижнюю часть документа.
4. Запишите формулу функции $r(x)$ для определения координат точек, лежащих на аппроксимирующей прямой. Коэффициенты соответствующего уравнения были получены в предыдущем упражнении.

$$r(x) := b_0 + b_1 \cdot x.$$

5. Нажмите клавишу @, щелкните на кнопке X-Y Plot (Декартовы координаты) на панели инструментов Graph (График) или дайте команду Insert ► Graph ► X-Y Plot (Вставка ► График ► Декартовы координаты). В документе появится область для создания графика.
6. Вместо заполнителя в нижней части графика укажите в качестве независимой переменной первый столбец матрицы data (data<0> или X).
7. Вместо заполнителя слева от графика укажите, что по вертикальной оси должны откладываться значения из второго столбца матрицы data и определенная выше линейная функция $r(X)$. В качестве разделителя используется запятая. Диапазон значений для осей координат выбирается программой Mathcad автоматически.
8. Чтобы изменить вид автоматически построенного графика, дважды щелкните внутри него. Откроется диалоговое окно Formatting Currently Selected X-Y Plot (Форматирование графика в декартовых координатах). Первая запись в списке на вкладке Traces (Кривые) соответствует первой отображенной кривой. Для изменения записи используются поля под списком.
9. В столбце Legend Label (Подпись) введите название графика.
10. В столбце Symbol (Маркер) выберите способ обозначения для отдельных точек.
11. В столбце Type (Вид линии) укажите, что необходимо пометить отдельные точки (points), а не провести непрерывную линию.
12. Выберите в списке вторую кривую и настройте ее отображение по своему вкусу.

13. Установите флажок Hide Arguments (Скрыть параметры), чтобы не отображать названия осей.
14. Сбросьте флажок Hide Legend (Скрыть подписи), чтобы включить отображение под графиком заданных подписей кривых.

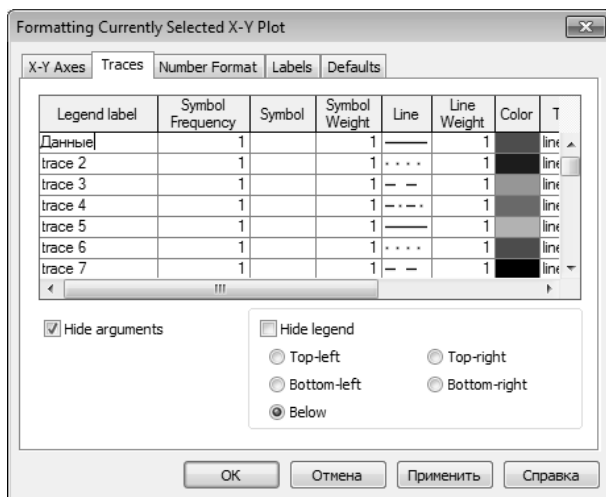


Рис. 18.5. Задание способа отображения линий графика

15. В поле Title (Заголовок) на вкладке Labels (Надписи) задайте название графика и включите режим его отображения: флажок Show Title (Показать заголовок).
16. Постройте график, на котором отображалась бы величина отклонения экспериментальных точек от линии приближения $Y - (b_0 + b_1 \cdot X)$. Отформатируйте его, используя те же средства, что и в предыдущем случае.

Итог. Заголовок и подписи, использующие русские буквы, могут отображаться неправильно. Коррекцию обеспечивает выбор шрифта, правильно воспроизводящего кириллицу. Дайте команду Format ► Equation (Формат ► Выражение), в раскрывающемся списке Style Name (Имя стиля) выберите пункт Variables (Переменные) и щелкните на кнопке Modify (Изменить). Для задания шрифта используйте поле со списком Шрифт.

Упражнение 18.7. Построение трехмерных графиков

Задача. Изобразить на графике приблизительную форму электронных облаков в атомах.

Анализ. По современным представлениям, электронные уровни в атоме определяются четырьмя квантовыми числами. Форма электронного облака определяется двумя из этих чисел:

- число l определяет тип орбитали (значения 0–3 соответствуют s -, p -, d - и f -орбиталам);
- число m определяет магнитный момент электрона и может изменяться в диапазоне от $-l$ до l .

При $m = 0$ форма электронного облака определяется на основе многочленов Лежандра первого рода:

$$P(x) = \frac{1}{2^l \cdot l!} \cdot \frac{d^l}{dx^l} (x^2 - 1)^l, \text{ где } l - \text{степень многочлена.}$$

$$\text{В этом случае } Y(\varphi) = \sqrt{\frac{2l+1}{4\pi}} \cdot |P(\cos \varphi)|.$$

Параметрическое задание соответствующей поверхности имеет следующий вид:

$$x(\theta, \phi) = Y(\phi) \cdot \sin \phi \cdot \cos \theta$$

$$y(\theta, \phi) = Y(\phi) \cdot \sin \phi \cdot \sin \theta$$

$$z(\theta, \phi) = Y(\phi) \cdot \cos \phi$$

Углы θ, ϕ изменяются в диапазоне от 0 до 2π .

1. Запустите программу Mathcad.
2. Определите переменную l , которая укажет тип орбитали.

$$l := 3$$

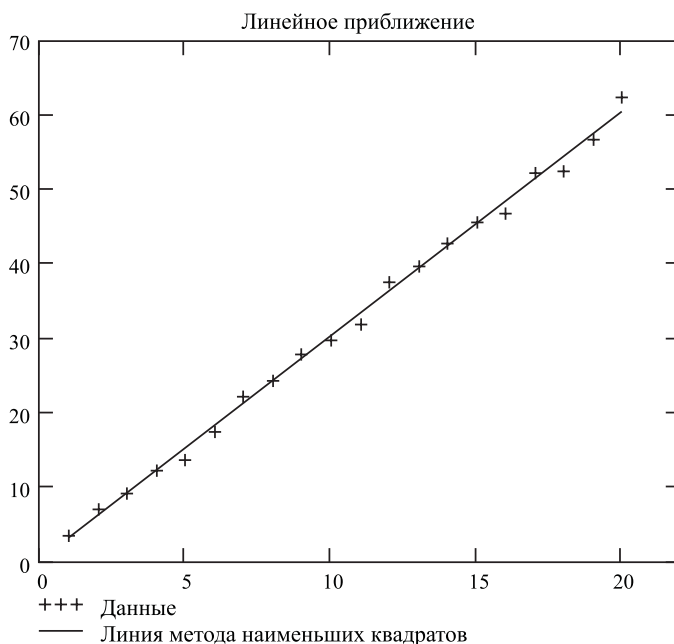


Рис. 18.6. График экспериментальных точек и аппроксимирующей прямой

Построение поверхности будем производить по точкам. Задайте два диапазона, которые будут определять изменение параметров θ, ϕ , задающих поверхность. Удобно определить границы диапазона в целых числах (через точку

с запятой, на экране изображаются две точки), а затем произвести перемасштабирование на отрезок $[0; 2\pi]$.

$$i := 0..100 \quad j := 0..100;$$

$$\theta_1 = i \cdot \frac{2 \cdot \pi}{100}, \quad \varphi_j = j \cdot \frac{2 \cdot \pi}{100}.$$

3. Определите двумерные матрицы, определяющие значения координат x , y и z в зависимости от значения параметров. Используйте названия переменных X_0 , Y_0 и Z_0 .

$$P(x) = \frac{1}{2^l \cdot l!} \cdot \frac{d^l}{dx^l} (x^2 - 1)^l;$$

$$Y(\phi) = \left| \sqrt{\frac{2l+1}{4\pi}} \cdot P(\cos \phi) \right|;$$

$$X_{0ij} = Y(\phi_j) \cdot \sin(\phi_j) \cdot \cos(\theta_i);$$

$$Y_{0ij} = Y(\phi_j) \cdot \sin(\phi_j) \cdot \sin(\theta_i);$$

$$Z_{0ij} = Y(\phi_j) \cdot \cos(\phi_j).$$

4. Дайте команду Insert ► Graph ► Surface Plot (Вставка ► График ► Поверхность) или воспользуйтесь кнопкой Surface Plot (Поверхность) на панели инструментов Graph (График).
5. В появившейся области графика вместо заполнителя укажите имена отображаемых матриц через запятую, заключив все их в скобки: (X0,Y0,Z0).
6. Чтобы изменить формат построенного графика, дважды щелкните на его области. Откроется диалоговое окно 3-D Plot Format (Формат трехмерного графика).
7. На вкладке General (Общие) установите флажок Equal Scales (Равный масштаб), чтобы обеспечить одинаковый масштаб по осям координат.
8. На вкладке Appearance (Оформление) установите переключатель Fill Surface (Заливка поверхности), чтобы обеспечить заливку построенной поверхности.
9. На вкладке Lighting (Подсветка) включите режим освещения поверхности. Установите флажок Enable Lighting (Включить подсветку), отключите все источники света, кроме первого.
10. На панели Light Direction (Направление света) задайте координаты источника света. Используйте кнопку Применить, чтобы сразу видеть последствия сделанных настроек. По окончании настройки закройте диалоговое окно щелчком на кнопке ОК.
11. Путем протягивания мыши в области графика измените направление осей координат, чтобы изображение было видно наиболее отчетливо.
12. Изменяя значение l , можно увидеть форму электронных облаков для разных орбиталей, в том числе и не встречающихся в природе.

Итог. Мы научились строить трехмерные графики с изображением объемных поверхностей, заданных параметрически. Это фактически означает умение изображать любые фигуры, которые могут потребоваться в ходе практической работы.

Упражнение 18.8. Решение дифференциальных уравнений

Задача. Найти функцию $y(x)$, удовлетворяющую дифференциальному уравнению $\frac{dy}{dx} + y = x \cdot \cos x$ и имеющую значение 0 при $x = 0$.

Анализ. Это простое дифференциальное уравнение допускает точное аналитическое решение. В данном упражнении предполагается использование стандартной функции программы Mathcad, осуществляющей численное решение данного уравнения. Результат вычислений можно после этого сравнить с точным решением.

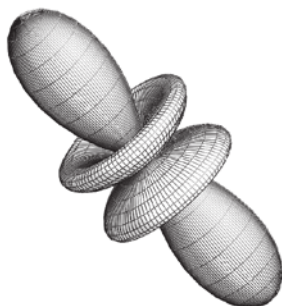


Рис. 18.7. Трехмерное изображение электронной f-орбитали

1. Запустите программу Mathcad.
2. Задайте начальное значение функции как элемент вектора y , размерность которого соответствует числу решаемых уравнений (в данном случае единице): $y_0 := 0$.
3. Создайте функцию $T(x, y)$, которая вычисляет значение производной при заданных значениях независимой переменной и неизвестной функции:

$$T(x, y) := -y_0 + x \cdot \cos(x).$$

4. Определите начальное (точка 0) и конечное значение отрезка интегрирования.

$$a := 0, \quad b := 12 \cdot \pi.$$

5. Укажите число шагов интегрирования.

$$K := 20.$$

6. Вычислите численное решение уравнения при помощи функции *rkfixed*.

$$Z := rkfixed(y, a, b, K, T).$$

7. Результат вычислений — матрица Z с двумя столбцами, первый из которых содержит значения независимой переменной, а второй — соответствующие значения функции.
8. Постройте график полученного решения.
9. Определите аналитическое решение данного уравнения при тех же начальных условиях.
10. Нанесите аналитическую кривую на тот же график и сравните поведение численного и точного решения.
11. Измените число шагов, на которые делится отрезок интегрирования, и исследуйте, как изменится результат расчета при уменьшении и увеличении этого параметра.

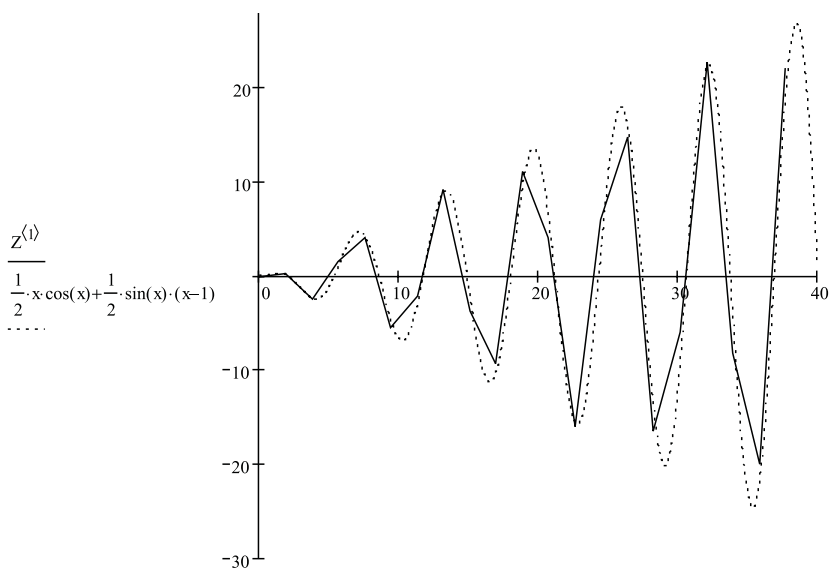


Рис. 18.8. Графики численного и точного решения дифференциального уравнения

Итог. Мы научились численно решать дифференциальные уравнения первого порядка с помощью программы Mathcad. Использованный метод без изменений переносится на системы, содержащие два или большее число дифференциальных уравнений. Увеличение величины шага интегрирования ускоряет получение результата, но снижает его точность. При слишком большой величине шага результат расчетов может вообще не соответствовать реальному решению.

Глава девятнадцатая

Публикация веб-документов

Размещение собственных материалов в Интернете включает два этапа: *подготовку* материалов и их *публикацию*. Подготовка материалов состоит в создании документов, имеющих формат, принятый в Интернете, то есть веб-страниц, написанных на языке HTML. Публикация материалов, то есть открытие к ним доступа, осуществляется после решения организационных вопросов, связанных с получением дискового пространства на веб-сервере для их размещения.

19.1. Создание веб-документов

Автономные веб-документы используют язык *HTML (HyperText Markup Language — язык разметки гипертекста)*. *Гипертекст*, то есть *расширенный текст*, включает дополнительные элементы: иллюстрации, ссылки, вставные объекты. Под *разметкой* понимается использование специальных кодов, легко отделяемых от смыслового содержания документа и используемых для реализации гипертекста. Применение этих кодов подчиняется строгим правилам, определяемым спецификацией языка HTML.

Особенность описания документа средствами языка HTML связана с *принципиальной невозможностью достижения абсолютной точности воспроизведения исходного документа*. Предполагается, что документ будет широко доступен в Интернете, и поэтому *неизвестно, как будет организовано его воспроизведение*. Документ может быть представлен на графическом экране, выведен в чисто текстовом виде или просто «прочитан» программой синтеза речи. Разметка HTML во всех этих случаях должна быть принята во внимание. Поэтому язык HTML предназначен не для *форматирования* документа, а для его *функциональной разметки*. Например, документы обычно начинаются с заголовков. Свойство части документа «быть заголовком» — это не особенность форматирования документа, а характеристика его содержания. Конкретное средство отображения документа (браузер) выбирает свой способ представления части документа, описанной как заголовок.

Современная версия HTML 4.01 (в настоящее время ведется работа над HTML 5) может в настоящий момент рассматриваться как «окончательная» редакция языка HTML, содержащая все необходимое для функциональной разметки документа. Недостаток оформительских средств и средств обеспечения интерактивности восполняется внешними по отношению к HTML средствами, такими как *списки стиля* и *динамические сценарии*. Многие веб-узлы выполняют *автоматическую генерацию* веб-страниц на основе содержания некоторой базы данных и запроса пользователя. Подобные элементы стали стандартными компонентами современных веб-страниц, но лежат за пределами данного пособия.

Управляющие конструкции языка HTML называются *тегами* и вставляются непосредственно в текст документа. Все теги заключаются в угловые скобки <...>. Сразу после открывающей скобки помещается ключевое слово, определяющее тег,

например <DIV>. Теги HTML бывают *парными* и *непарными*. *Непарные* теги оказывают воздействие на весь документ или определяют разовый эффект в месте своего появления. При использовании парных тегов в документ добавляются *открывающий* и *закрывающий* теги, которые воздействуют на часть документа, заключенную между ними. Закрывающий тег отличается от открывающего наличием символа «/» (косая черта) перед ключевым словом (</DIV>). Закрытие парных тегов выполняется так, чтобы соблюдались правила вложения.

```
<B><I>На этот текст воздействуют два тега</I></B>
```

Эффект применения тега может видоизменяться путем добавления *атрибутов*. В парных тегах атрибуты добавляются *только к открывающему тегу*. Атрибуты представляют собой дополнительные ключевые слова, отделяемые от ключевого слова, определяющего тег, и от других атрибутов пробелами и размещаемые до завершающего тег символа «>». Способ применения некоторых атрибутов требует указания *значения атрибута*. Значение атрибута отделяется от ключевого слова атрибута символом «=» (знак равенства) и заключается в кавычки.

```
<H1 ALIGN="LEFT">
```

Определение HTML как *языка разметки* основывается на том, что при удалении из документа всех тегов получается *текстовый* документ, совершенно эквивалентный по содержанию исходному *гипертекстовому* документу. Таким образом, при отображении документа HTML сами теги *не отображаются*, но влияют на способ отображения остальной части документа.

Если говорить о создании документов HTML, то можно представить себе два способа их формирования. Первый состоит в разметке существующего (или создаваемого) документа *вручную*. При этом автор или редактор добавляет в документ теги разметки. Эту работу можно выполнять в *текстовом редакторе* или *редакторе HTML*, имеющем специальные элементы управления для упрощения ввода тегов. В обоих этих случаях работа ведется средствами языка HTML, и человек, выполняющий эту работу, должен знать и уметь применять этот язык.

Принципы иного подхода можно понять на основе изучения работы текстовых процессоров. Информацию о форматировании документа можно рассматривать как «разметку», добавляемую в формируемый документ. Однако для использования текстового процессора не требуется знаний о формате документа и «языке разметки»: изменения, отображаемые на экране, вносятся в документ автоматически. Такой принцип соответствия экранного изображения реальному получил название WYSIWYG (от английского *What You See Is What You Get* — что видите, то и получаете).

В качестве редактора WYSIWYG для языка HTML можно использовать текстовый процессор Word или входящую в состав пакета Microsoft Visual Studio программу Microsoft Visual Web Developer. Существуют и другие программы того же самого назначения.

Работа вручную позволяет создавать более универсальные, более качественные и более разнообразные документы. Второй способ проще освоить, так как он не требует знания языка HTML. Однако в этом случае используются средства *форматирования* вместо средств *описания*, что может иногда приводить к нежелательным последствиям.

При подготовке крупных веб-узлов, содержащих десятки, а то и сотни веб-страниц, та или иная автоматизация работы необходима. Как правило, в таких случаях используют комбинированный подход: «рядовые» страницы готовят с помощью автоматизированных средств, а в особо сложных или в особо важных случаях задают оформление вручную.

Процесс создания веб-документов сродни программированию и так же подвержен ошибкам. Независимо от того, каким способом создается документ, следует регулярно проверять его соответствие замыслу, просматривая его в различных браузерах. Для художественной оценки получающейся страницы следует обратиться к независимому мнению.

19.2. Применение языка HTML

Структура документа HTML

Все документы HTML имеют одну и ту же структуру, определяемую фиксированным набором *тегов структуры*. Документ HTML всегда должен начинаться с тега `<HTML>` и заканчиваться соответствующим закрывающим тегом (`</HTML>`). Внутри документа выделяются два основных раздела: *раздел заголовков* и *тело документа*, идущих именно в таком порядке. Раздел заголовков содержит информацию, описывающую документ в целом, и ограничивается тегами `<HEAD>` и `</HEAD>`. В частности, раздел заголовков должен содержать общий заголовок документа, ограниченный парным тегом `<TITLE>`.

Основное содержание размещается в теле документа, которое ограничивается парным тегом `<BODY>`. Строго говоря, положение структурных тегов в документе нетрудно определить, даже если они опущены. Поэтому стандарт языка HTML *требует* только наличия тега `<TITLE>` (и, соответственно, `</TITLE>`). Тем не менее при создании документа HTML опускать структурные теги не рекомендуется.

Простейший правильный документ HTML, содержащий все теги, определяющие структуру, может выглядеть следующим образом:

```
<HTML>
<HEAD><TITLE>Заголовок документа</TITLE></HEAD>
<BODY>
Текст документа
</BODY>
</HTML>
```

Элементы HTML

Для парных тегов область влияния определяется частью документа между открывающим и закрывающим тегом. Такую часть документа рассматривают как *элемент языка HTML*. Так, можно говорить об «элементе `BODY`», включающем тег `<BODY>`, основное содержание документа и закрывающий тег `</BODY>`. Весь документ HTML можно рассматривать как «элемент HTML». Для непарных тегов элемент совпадает с тегом, который его определяет.

Большинство элементов языка HTML описывает части содержания документа и помещается между тегами `<BODY>` и `</BODY>`, то есть внутри структурного эле-

мента BODY. Такие элементы делят на *блочные* и *текстовые*. Блочные элементы относятся к частям текста уровня *абзаца*. Текстовые элементы описывают свойства *отдельных фраз* и еще более мелких частей текста.

Теперь можно сформулировать *правила вложения элементов*.

- ❑ **Элементы не должны пересекаться.** Другими словами, если открывающий тег располагается внутри элемента, то и соответствующий закрывающий тег должен располагаться внутри этого же элемента.
- ❑ **Блочные элементы могут содержать вложенные блочные и текстовые элементы.**
- ❑ **Текстовые элементы могут содержать вложенные текстовые элементы.**
- ❑ **Текстовые элементы не могут содержать вложенные блочные элементы.**

Строго говоря, все правила языка HTML можно рассматривать исключительно как «пожелания». Средство, используемое для отображения веб-документа, делает все возможное, чтобы истолковать разметку наиболее разумным образом. Тем не менее *гарантию* правильного воспроизведения документа дает только неукоснительное следование требованиям спецификации языка.

Функциональные блочные элементы

В большинстве документов основными функциональными элементами являются *заголовки* и *абзацы* (рис. 19.1). Язык HTML поддерживает шесть уровней заголовков. Они задаются при помощи парных тегов от <H1> до <H6>. При отображении веб-документа на экране компьютера эти элементы показываются при помощи шрифтов разного размера.

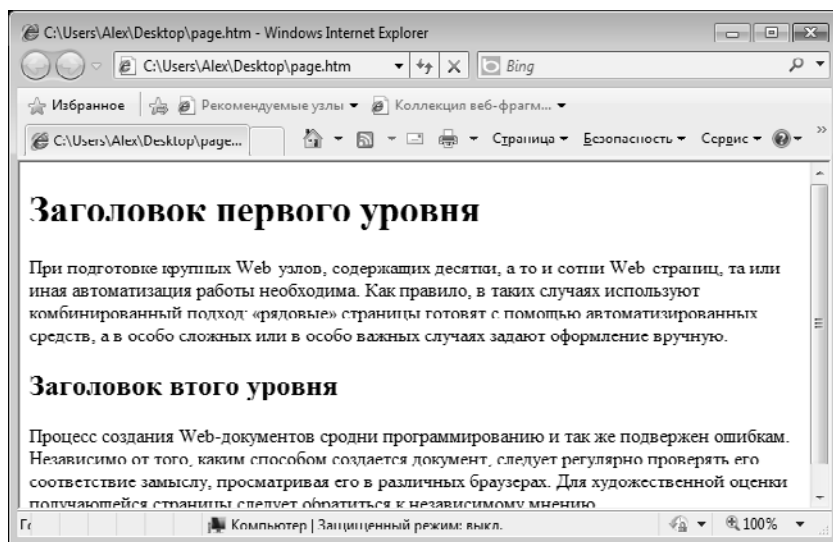


Рис. 19.1. Форматирование заголовков и абзацев при отображении веб-страницы в окне браузера

Обычные абзацы задаются с помощью парного тега `<P>`. Язык HTML не содержит средств для создания абзацного отступа («красной строки»), поэтому при отображении на экране компьютера абзацы разделяются пустой строкой. Закрывающий тег `</P>` рассматривается как *необязательный*. Подразумевается, что он стоит перед тегом, который задает начало очередного абзаца документа. Например:

```
<H1>Заголовок</H1>
<P>Первый абзац<P>Второй абзац
<H2>Заголовок второго уровня</H2>
```

Следствием наличия специального тега, определяющего абзац, является тот факт, что обычного символа конца строки, вводимого по нажатию клавиши `ENTER`, для создания абзацного отступа недостаточно. Язык HTML рассматривает символы конца строки и пробелы особым образом. Любая последовательность, состоящая только из пробелов и символов конца строки, при отображении документа рассматривается как одиночный пробел. Это, в частности, означает, что символ конца строки даже не осуществляет перехода на новую строку (для этой цели используется текстовый элемент, задаваемый непарным тегом `<BR>`).

В качестве ограничителя абзацев может также использоваться *горизонтальная линейка*. Этот элемент задается непарным тегом `<HR>`. При отображении документа на экране линейка разделяет части текста друг от друга. Ее длина и толщина задается атрибутами тега `<HR>`.

```
<HR ALIGN="RIGHT" SIZE="10" WIDTH="50%">
```

Этот тег создает горизонтальную линейку шириной в 10 пикселей, занимающую половину ширины окна и расположенную справа.

Гипертекстовые ссылки

Гипертекстовая ссылка (рис. 19.2) является фрагментом текста документа и потому задается текстовым элементом, определяемым при помощи парного тега `<A>`. Этот элемент содержит обязательный атрибут, который не может быть опущен. В данном случае обязательным является атрибут `HREF=` (знак равенства показывает, что необходимо задать *значение* этого атрибута).

В качестве значения атрибута используется адрес URL документа, на который указывает ссылка. Она может указывать на произвольный документ, располагающийся на любом общедоступном узле сети (веб-узел, архив *FTP* и пр.). Например, открывающий тег ссылки может иметь вид `<A HREF="http://www.site.com/index.htm">`.

Адрес URL может быть задан в *абсолютной форме*, то есть начинаться с указания протокола и адреса веб-узла. Такая запись адреса используется, когда необходимо направить посетителя на другой веб-узел, и рассматривается как *внешняя ссылка*. При использовании *относительного адреса* в ссылке задается только относительный путь поиска для документа. В этом случае предполагается использование того же протокола и того же веб-узла, а ссылка рассматривается как *внутренняя*. Внутренняя ссылка сохраняет свою работоспособность в случае изменения адреса веб-узла как целого (например, в результате его переноса на другой сервер), поэтому при потенциальной возможности такого события следует отказываться от полного задания адресов в гиперссылках.

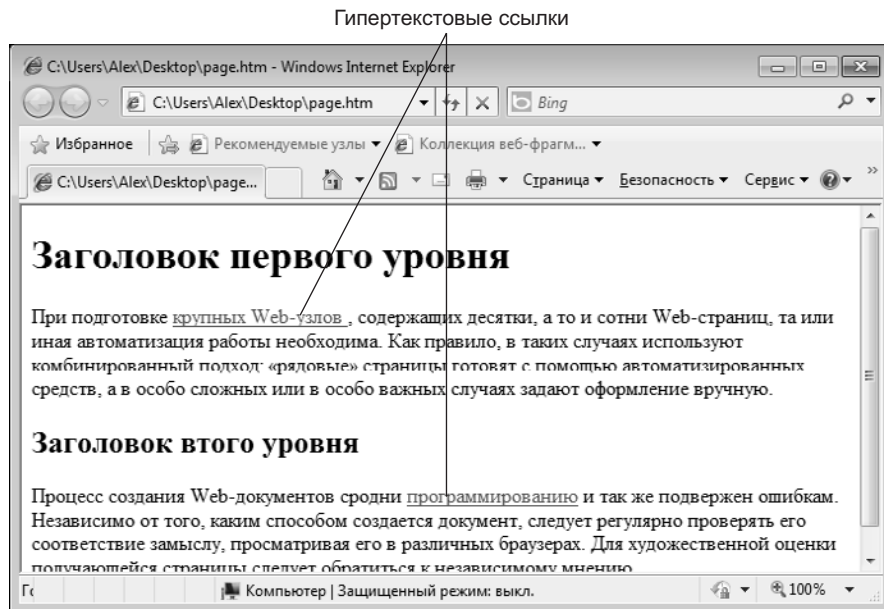


Рис. 19.2. На веб-странице гипертекстовые ссылки выделяются цветом и подчеркиванием

Гиперссылки можно использовать для ссылки на мультимедийные файлы. Это удобно, так как в этом случае не приходится ждать загрузки мультимедийных файлов при работе с данной страницей. Если же требуется интегрирование объектов мультимедиа в веб-страницу, используют парный тег `<OBJECT>` или нестандартный непарный тег `<EMBED>`, который тоже поддерживается наиболее распространенными браузерами.

Полный формат гиперссылки включает возможность ссылки на *определенное место внутри страницы*. Но это можно сделать для страниц собственной разработки, пометив соответствующее место при помощи якоря. Якорь задается также при помощи парного тега `<A>`, но в роли обязательного выступает атрибут `NAME=`. Значение этого атрибута — произвольная последовательность латинских букв и цифр (пробелы недопустимы), рассматриваемая как *имя якоря*. Для ссылки на якорь его имя указывается в конце адреса URL после символа `#`.

```
<A HREF="http://www.site.com/index.htm#address">
```

Веб-графика

Графические иллюстрации в большинстве случаев являются неотъемлемой частью веб-документов. Сегодня графические элементы веб-страниц используют два основных формата — GIF и JPEG (допустим также формат PNG, который, однако, так и не получил широкого распространения). Все графические браузеры, предназначенные для отображения веб-страниц на экране компьютера, способны распознавать и отображать файлы этих форматов.

Для подготовки изображений можно использовать любой графический редактор, например стандартное приложение Paint (в Windows XP), которое позволяет сохранять файлы в этих форматах.

Файлы формата GIF (Graphic Interchange Format) имеют расширение .GIF. Изображения в этом формате содержат 256 цветов, заданных индексной палитрой. Файл упакован и может занимать значительно меньше места, чем неупакованный растровый рисунок (например, в формате .BMP).

Спецификация формата GIF89a позволяет создавать файлы .GIF, обладающие специальными возможностями.

- ❑ Один из цветов изображения может быть объявлен *прозрачным*. Это означает, что в соответствующих местах сквозь него будет проглядывать фон веб-страницы, что позволяет задать не только прямоугольную форму рисунка и делает его более естественным.
- ❑ *Чересстрочные изображения* при их приеме из Интернета прорисовываются постепенно, вначале грубо, а затем все более и более четко. Это «скрадывает» время, необходимое на их загрузку из Интернета, особенно при приеме информации по медленным линиям.
- ❑ *GIF-анимация* превращает обычный рисунок в небольшой видеоролик. В стандартном файле с расширением .GIF хранится набор кадров, а также сценарий их отображения.

Для создания файлов .GIF, использующих эти расширенные возможности, необходим графический редактор, более мощный, чем программа Paint. Для создания GIF-анимации используют специальные средства.

Файлы формата JPEG (Joint Photographic Expert Group — по названию группы исследователей, предложившей этот формат, читается «джей-пег») могут иметь расширение .JPEG или .JPG. Формат предназначен для хранения фотографических изображений, использующих 24-разрядный цвет. При конвертировании в формат JPEG происходит потеря части информации, приводящая к некоторому ухудшению качества изображения, обычно незаметному на глаз.

При выборе формата изображения в первую очередь принимают во внимание объем получающегося файла и во вторую — качество изображения. При загрузке веб-документа львиную долю времени занимает именно загрузка иллюстраций, так что любая экономия приветствуется. При выборе формата рекомендуется создать два файла: в формате GIF и в формате JPEG с минимально приемлемым качеством, после чего выбрать вариант, имеющий меньший объем.

Рисунки хранятся на веб-узлах в отдельных файлах, но отображаются как элементы веб-страниц. Для вставки рисунка используется текстовый элемент, задаваемый непарным тегом . Тег должен содержать обязательный атрибут SRC=, задающий адрес URL файла с изображением в относительной или абсолютной форме.

```
<IMG SRC="picture1.gif">
```

При отображении рисунка браузер по умолчанию использует его реальные размеры. Если рисунок необходимо отмасштабировать, применяют атрибуты WIDTH= и HEIGHT=, задающие ширину и высоту рисунка (в пикселах) соответственно. Если

эти параметры заданы, то браузер может определить, какое место надо выделить для отображения рисунка, еще до того, как рисунок загружен. Это несколько ускоряет отображение загружаемой страницы, так что удобно задавать эти атрибуты всегда.

```
<IMG SRC="picture2.jpg" WIDTH="100" HEIGHT="40">
```

Внешний вид веб-страницы зависит от того, как именно рисунок располагается на ней. Так как рисунок задается как текстовый элемент, находящийся внутри какого-то абзаца, по умолчанию он рассматривается как *встроенное изображение*, включаемое в строку текста. Чтобы изображение отображалось автономно, его включают в отдельный абзац.

Для изображения, которое действительно включено в строку, можно задать режим взаимодействия с текстом с помощью атрибута **ALIGN**=.

```
<IMG SRC="picture3.gif" ALIGN="BOTTOM">
```

Этот атрибут может принимать три значения:

- ☐ если задано **ALIGN="BOTTOM"**, то нижняя граница изображения совмещается с основанием текстовой строки;
- ☐ если задано **ALIGN="MIDDLE"**, то середина изображения совмещается с серединой текстовой строки;
- ☐ если задано **ALIGN="TOP"**, то верхняя граница изображения выравнивается по верхнему обрезу текстовой строки.

Однако более предпочтительно использование «плавающего» изображения, обтекаемого текстом, что также достигается использованием атрибута **ALIGN**=:

- ☐ если задано **ALIGN="LEFT"**, то изображение размещается у левого края страницы, а последующий текст размещается справа от него;
- ☐ если задано **ALIGN="RIGHT"**, то изображение размещается у правого края страницы, а последующий текст размещается слева от него.

В этом случае рекомендуется помещать тег **** в самое начало соответствующего абзаца.

Однако нормальный режим обтекания требует, чтобы между текстом и изображением оставался некоторый промежуток (рис. 19.3). Задать величину этого промежутка можно при помощи атрибутов **HSPACE**= (по горизонтали) и **VSPACE**= (по вертикали). Размеры задаются в пикселах.

Создавая иллюстрированные страницы, не следует забывать, что не все смогут увидеть эти иллюстрации. Читателей, не имеющих адекватного средства просмотра, можно ознакомить с содержанием иллюстраций при помощи *альтернативного текста*. Альтернативный текст задается как значение атрибута **ALT**= и отображается вместо картинки, если она по каким-то причинам не может быть выведена.

Так как изображение задается как текстовый элемент, оно может быть помещено внутрь другого текстового элемента, например задающего гиперссылку (тег **<A>**). В этом случае изображение становится *изображением-ссылкой*. При отображении документа на экране компьютера такое изображение отличается синей рамкой и изменением формы указателя при наведении.

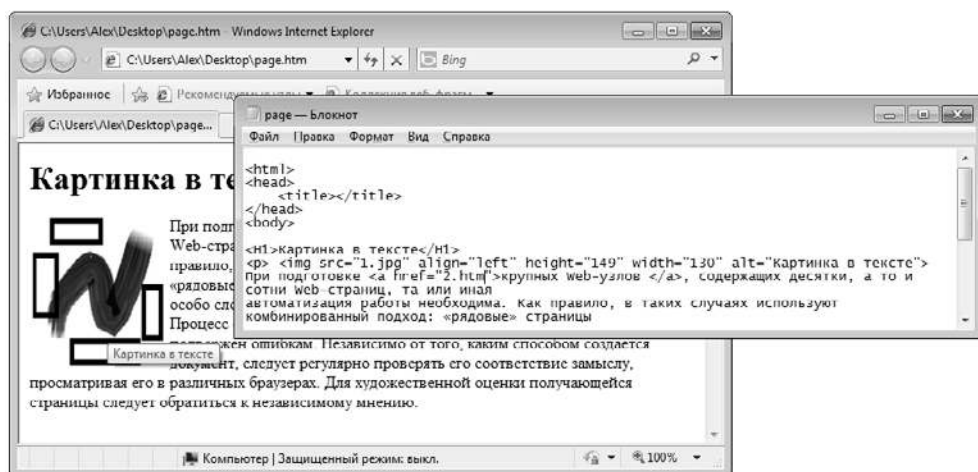


Рис. 19.3. Изображение в тексте, выровненное по левому полю

Еще один способ применения изображений на веб-страницах состоит в использовании их в качестве *фоновой рисунка*. При отображении документа, содержащего фоновый рисунок, на компьютере рабочая область окна заполняется этим рисунком последовательно (как паркетом), считая от верхнего левого угла документа или окна (рис. 19.4). Фоновый рисунок задается с помощью атрибута **BACKGROUND=** в теге **<BODY>**. Значением этого атрибута должен быть абсолютный или относительный адрес URL для файла с изображением.

```
<BODY BACKGROUND="waves.gif" TEXT="YELLOW">
```

Форматирование текста

Управление форматированием текста не является основной задачей языка HTML, и поэтому текстовые элементы, выполняющие эту задачу, начиная с версии 4.0 рассматриваются как устаревшие и их использование не рекомендуется. Однако они все еще предоставляют удобный способ управления видом документа на экране компьютера.

Парный тег **** позволяет управлять параметрами шрифта. Он должен обязательно содержать хотя бы один из трех атрибутов: **COLOR=**, **FACE=** или **SIZE=**.

```
<FONT SIZE="6">
```

Атрибут **COLOR=** задает цвет текста, который может быть задан текстовым значением (например, **COLOR="GREEN"**) или шестнадцатеричным кодом, в котором последовательные байты задают значения красной, зеленой и синей составляющих цвета (**COLOR="#00FF00"** дает тот же результат, что и **COLOR="GREEN"**). Атрибут **FACE=** задает гарнитуру шрифта. Значение этого атрибута сравнивается с именами шрифтов, которые установлены на компьютере. Атрибут **SIZE=** определяет размер

шрифта в относительных единицах (от 1 до 7). Для этого атрибута можно определять значение со знаком (плюс или минус), которое определяет увеличение или уменьшение шрифта относительно текущего размера.

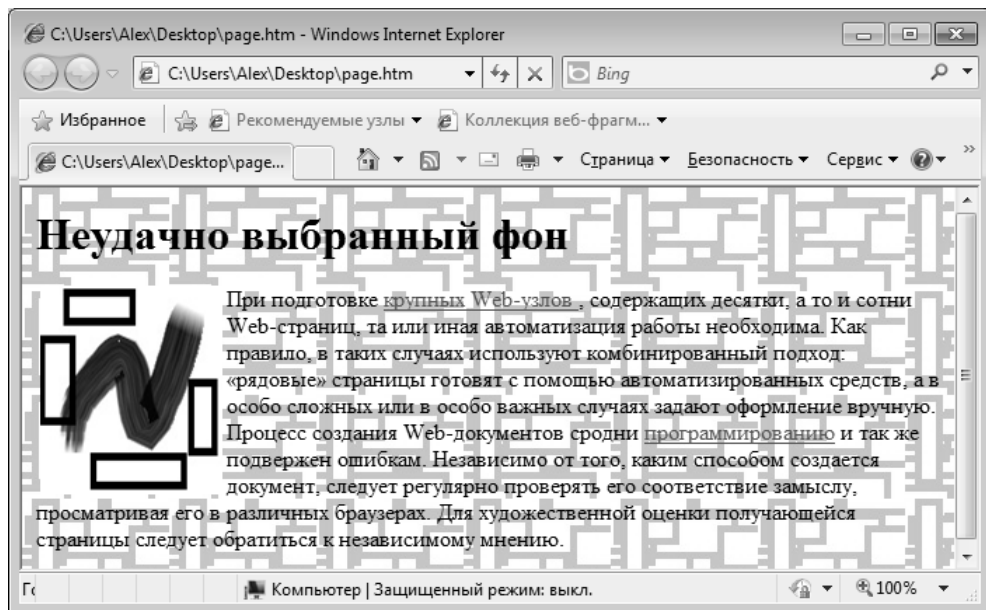


Рис. 19.4. К выбору фоновой рисунка следует подходить с особой осторожностью, так как неудачный фон может сильно затруднить чтение документа

Параметры шрифта, используемые в документе по умолчанию, задают с помощью непарного тега `<BASEFONT>`, который помещают один раз внутри элемента `BODY`. Он может использовать те же атрибуты, что и тег `<FONT>`.

Начертание символов задается при помощи парных тегов `<B>` (полужирный шрифт), `<I>` (курсив), `<U>` (подчеркнутый текст), `<S>` (вычеркнутый текст). Их использование не рекомендуется. Вместо них следует применять *элементы фразы*, описывающие функциональные особенности текста; например, вместо

```
<B>Обратите внимание!</B>
```

лучше написать

```
<STRONG>Обратите внимание!</STRONG>
```

Так, парный тег `<CITE>` предназначен для отображения *цитат* (выводятся курсивом). Парные теги `<EM>` (выделение) и `<STRONG>` (сильное выделение) являются функциональными аналогами курсивного и полужирного начертаний. Кроме того, язык HTML содержит набор элементов для описания работы компьютерных программ. Для этой цели используют парные теги `<CODE>` (исходный текст программы), `<KBD>` (текст, вводимый с клавиатуры), `<SAMP>` (пример вывода программы) и `<VAR>` (программные переменные). Для вывода соответствующих эле-

ментов используется моноширинный шрифт. Кроме того, переменные выводятся курсивом, а клавиатурный ввод (в некоторых браузерах) — полужирным шрифтом.

Списки

Язык HTML поддерживает пять видов списков, из которых два (*списки меню* и *списки каталогов*) считаются устаревшими и не рекомендуются к применению. Оставшиеся три типа — это *упорядоченные списки*, *неупорядоченные списки* и *списки определений*. Все списки представляют собой блочные элементы.

Упорядоченные (*нумерованные*) и неупорядоченные (*маркированные*) списки, примеры которых приведены на рис. 19.5, оформляются одинаково. Они создаются при помощи парных тегов: `<OL>` для упорядоченного списка и `<UL>` для неупорядоченного. Эти списки могут содержать только *элементы списка*, определяемые парным тегом `<LI>`. Закрывающий тег `</LI>` можно опускать, так как его местонахождение легко восстановить. Открывающие теги могут содержать атрибуты, определяющие вид маркера (для неупорядоченного списка), способ и последовательность нумерации (для упорядоченного). Разрешается вложение списков друг в друга.

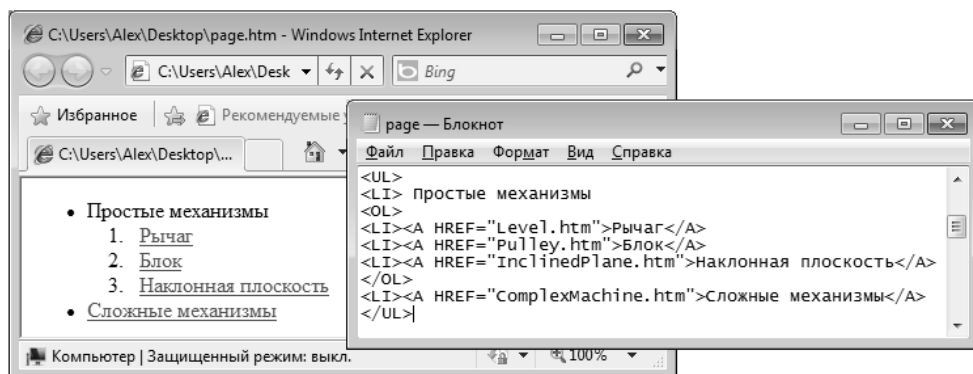


Рис. 19.5. Создание маркированных и нумерованных списков

Список определений задается парным тегом `<DL>`. Он содержит элементы двух типов: *определяемые термины* (парный тег `<DT>`) и *определения* (парный тег `<DD>`). Закрывающие теги `</DT>` и `</DD>` можно опускать. Обычно определяемые термины и определения чередуют, хотя это нигде не оговорено. Определения отображаются на экране с отступом от левого края. Такой список может быть сформирован следующим образом:

```
<DL>
<DT>Поршень
<DD>Сплошной цилиндр или диск, который плотно входит внутрь полого цилиндра
</DL>
```

Таблицы

Таблицы удобны для представления больших объемов данных, а многие веб-дизайнеры используют их также для точного размещения элементов веб-страниц (рис. 19.6).

Таблица в языке HTML задается при помощи парного тега `<TABLE>`. Она может содержать *заголовок таблицы*, определяемый парным тегом `<CAPTION>`, и *строки таблицы*, задаваемые при помощи парных тегов `<TR>`. Закрывающие теги `</TR>` можно опускать.

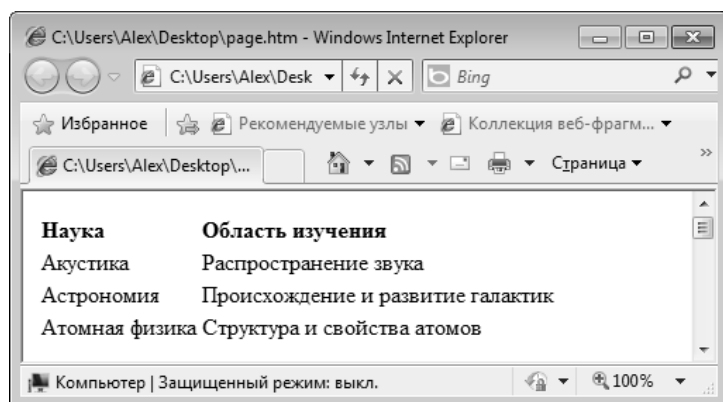


Рис. 19.6. Для размещения текста в двух колонках использована «невидимая» таблица

Каждая строка таблицы содержит *ячейки таблицы*, которые могут относиться к двум разным типам. Ячейки в заголовках столбцов и строк задают парным тегом `<TH>`, а обычные ячейки — парным тегом `<TD>`. Закрывающие теги `</TH>` и `</TD>` можно опускать. Например, «пустая» таблица с двумя строками и двумя столбцами может быть задана следующим образом:

```
<TABLE>
<CAPTION>Пустая таблица</CAPTION>
<TR><TD><TD>
<TR><TD><TD>
</TABLE>
```

Каждая ячейка может содержать произвольный текст, а также любые теги HTML, допустимые в «теле» документа. В частности, ячейка таблицы может содержать вложенную таблицу или изображение.

При отображении таблицы на экране компьютера происходит ее автоматическое форматирование с подбором размеров ячеек в соответствии с объемом размещаемой информации и заданными атрибутами. Атрибуты элементов позволяют сколь угодно причудливо оформить таблицу по своему вкусу. В табл. 19.1 приведена краткая сводка допустимых атрибутов.

Таблица 19.1. Атрибуты элементов, используемых при создании таблиц

Атрибут	Элемент	Назначение
ALIGN=	Таблица, заголовок, строка, ячейка	Выравнивание таблицы по горизонтали; выравнивание данных по горизонтали; размещение заголовка над или под таблицей
VALIGN=	Строка, ячейка	Выравнивание по вертикали
WIDTH=	Таблица, ячейка	Минимальная ширина
HEIGHT=	Ячейка	Минимальная высота
COLSPAN=	Ячейка	Протяженность в несколько столбцов
ROWSPAN=	Ячейка	Протяженность в несколько строк
BGCOLOR=	Таблица, ячейка	Цвет фона
CELLSPACING=	Таблица	Зазор между ячейками
CELLPADDING=	Таблица	Зазор между содержимым ячейки и ее границей
BORDER=	Таблица	Отображение границ ячеек и внешней рамки таблицы

Отображение нескольких документов

Язык HTML позволяет в рамках одной веб-страницы отобразить несколько документов. Для этого страница должна быть разбита на несколько областей — *фреймов*. Разбиение страницы описывается документом HTML особого рода, структура которого отличается от обычной. Тело документа заменяется *описанием фреймов*, задаваемым парным тегом `<FRAMESET>`. Элемент `BODY` в таком документе отсутствует, а при наличии — игнорируется браузером.

Открывающий тег `<FRAMESET>` должен содержать обязательный атрибут `COLS=` или `ROWS=`, определяющий способ разбиения окна. В первом случае окно разбивается вертикальными линиями, во втором — горизонтальными. Если заданы оба атрибута, создается сетка фреймов. Значение любого из этих атрибутов — это перечисленные через запятую размеры отдельных фреймов.

```
<FRAMESET COLS="60%, 40%">
```

Значения могут быть заданы в пикселах или в процентах от ширины окна. Последняя область может быть определена с помощью символа «*», это означает, что ей выделяется все оставшееся пространство.

```
<FRAMESET ROWS="40%, 40%, *">
```

Между тегами `<FRAMESET>` и `</FRAMESET>` должно располагаться ровно столько элементов, сколько областей создано с помощью атрибутов `COLS=` и `ROWS=`. При этом могут использоваться дополнительные элементы `FRAMESET`, описывающие дальнейшее разбиение на подобласти еще меньшего размера, или непарные теги `<FRAME>`, определяющие способ использования области.

Тег `<FRAME>` должен содержать обязательный атрибут `SRC=`, с помощью которого указывается, какой документ первоначально загружается в соответствующую

область. Значение этого атрибута — абсолютный или относительный адрес URL нужного документа.

Среди прочих атрибутов выделяется атрибут **NAME=**, позволяющий задать «имя» созданной области в виде последовательности латинских букв и цифр, использованной как значение этого атрибута.

```
<FRAME SRC="text.htm" NAME="left">
```

Это имя можно использовать, чтобы загружать новые документы в ранее созданную область. Для этого в тег **<A>**, определяющий гиперссылку, необходимо добавить атрибут **TARGET=**, значение которого совпадает с ранее определенным именем области. При переходе по данной гиперссылке новый документ загрузится в указанный фрейм.

Например, предположим, что начальная страница веб-узла состоит из двух фреймов: слева располагается навигационная панель, а справа — текущая страница. Если правой области присвоено имя, используемое во всех ссылках, имеющихся в левой области, то щелчок на любой ссылке навигационной панели приведет к обновлению информации в соседней области, оставляя навигационную панель без изменений.

Интерактивные веб-страницы

Веб-страницы являются интерактивными по самой своей природе, связанной с использованием гиперссылок. Но это пассивная интерактивность, жестко заданная в рамках структуры веб-узла. Подлинная интерактивность, позволяющая получать от посетителей веб-страницы произвольные данные, достигается путем использования форм.

Форма на веб-странице представляет собой лишь набор *полей*, которые можно также рассматривать как *элементы управления* (рис. 19.7). Посетитель в процессе работы с веб-страницей заполняет форму, после чего отправляет ее. Далее поступившие данные обычно передают специальной программе, предназначенной для ее обработки (*программе или сценарию CGI*).

Форма — это блочный элемент, описываемый парным тегом **<FORM>**. Теги, задающие поля формы, можно использовать только внутри этого элемента. Открывающий тег **<FORM>** определяет способ обработки формы при помощи нескольких атрибутов. Атрибут **METHOD=** определяет способ передачи предоставленных пользователем данных. Он может иметь два значения. Значение **"GET"** указывает, что данные будут переданы программе (или сценарию) CGI. В этом случае атрибут **ACTION=** содержит адрес URL этой программы. Если указано **METHOD="POST"**, то данные, введенные пользователем, отправляются по электронной почте. Атрибут **ACTION=** в этом случае должен содержать нужный адрес электронной почты с указанием протокола **mailto:.** Кроме этих, можно также указать атрибут **ENCTYPE=**, значение которого определяет тип MIME для отправки информации по электронной почте. По умолчанию используется значение **application/x-www-form-urlencoded**, но если форма предназначена для обработки вручную, лучше использовать тип MIME **text/plain**.

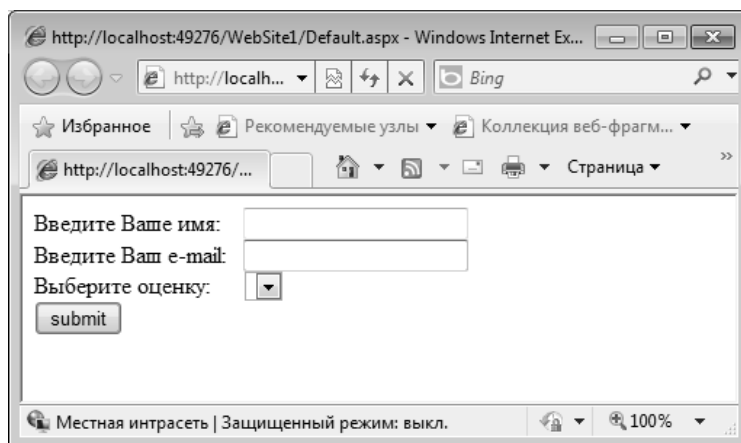


Рис. 19.7. Пример формы на веб-странице

Внутри элемента формы располагаются поля формы. Они задаются при помощи различных тегов. Вот те, которые используются чаще всего.

- ❑ Непарный тег `<INPUT>` позволяет создавать различные элементы управления, в том числе текстовые поля и командные кнопки.
- ❑ Парный тег `<TEXTAREA>` определяет текстовые области.
- ❑ Парный тег `<SELECT>` позволяет создавать обычные и раскрывающиеся списки `<OPTION>` (закрывающий тег `</OPTION>` можно опускать).

Тег `<INPUT>` должен содержать обязательный атрибут `TYPE=`, определяющий конкретный тип элемента управления. Вот основные возможные значения этого атрибута:

- ❑ `"TEXT"` — создается текстовое поле;
- ❑ `"PASSWD"` — создается текстовое поле, но вводимая информация не отображается на экране («текстовое поле для ввода пароля»);
- ❑ `"CHECKBOX"` — создается флажок, который может быть установлен или сброшен;
- ❑ `"RADIO"` — создается переключатель (из группы переключателей может быть включен только один);
- ❑ `"SUBMIT"` — создается *кнопка отправки* формы;
- ❑ `"IMAGE"` — создается графическая кнопка отправки;
- ❑ `"RESET"` — создается кнопка очистки формы, щелчок на которой возвращает форму к ее исходному состоянию.

Организация передачи данных, введенных в форму, осуществляется следующим образом. Каждый из тегов, задающих поля формы, должен иметь атрибуты `NAME=` и `VALUE=`. Атрибут `NAME=` определяет имя поля формы, атрибут `VALUE=` — значение поля. Для текстового поля и текстовой области атрибут `VALUE=` приобретает значение, соответствующее содержимому этого поля, заданному пользователем. Для элемента `SELECT` атрибут `VALUE=` соответствует содержимому выбранного

элемента **OPTION**. Все переключатели одной группы должны иметь одинаковые значения атрибута **NAME=** и разные значения атрибута **VALUE=**.

По щелчку на кнопке отправки данные из формы передаются в виде пар текущих значений атрибутов **NAME=** и **VALUE=**, соединенных знаком равенства. Информация о флажке передается только в том случае, если он установлен. Если атрибуты **NAME=** и **VALUE=** определены для использованной кнопки отправки, соответствующие данные также передаются (это позволяет включать в форму несколько кнопок отправки). Данные, поступающие в таком виде, удобны как для ручной, так и для автоматической обработки.

Создавая формы, следует иметь в виду, что информацию, передаваемую по электронной почте, нельзя считать конфиденциальной. Большинство браузеров может предупредить пользователя о возможности постороннего доступа к передаваемой информации. Эту особенность надо учитывать как при создании веб-страниц, содержащих формы, так и при заполнении форм на веб-страницах, встретившихся в Интернете.

19.3. Работа в Microsoft Visual Web Developer

Программа Microsoft Visual Web Developer входит в состав пакета Visual Studio. Она рассчитана на разработку и поддержку полноценных веб-узлов, в состав которых могут входить как обычные статичные HTML-страницы, так и динамические страницы, содержимое которых перед отправкой клиентскому приложению генерируется на сервере с использованием информации из серверной базы данных или из других источников.

Средства создания статичных страниц, рассматриваемые в этом пособии, — это всего лишь небольшая и не основная часть возможностей этой программы.

Начиная работу в Microsoft Visual Web Developer или в любой другой программе, рассчитанной на разработку веб-страниц, следует отдавать себе отчет, что в результате его применения получается документ HTML, построенный по тем же правилам, что и создаваемый вручную. Этот факт сразу же определяет возможности и ограничения в работе этого редактора.

- ☐ Все функции визуального редактора программы однозначно реализуются тегами HTML.
- ☐ Редактор не имеет средств, которые нельзя было бы представить в виде тегов HTML.
- ☐ Пользователь обычно не знает, какие именно средства HTML используются для достижения заданного эффекта и насколько корректно они применяются.

Создание и редактирование документа

Окно программы Microsoft Visual Web Developer (рис. 19.8) представляет собой комбинацию *окна редактора* и *окна браузера*. Документ HTML отображается редактором как специфическим браузером, отображающим даже обычно невидимые элементы (такие, как якоря). В то же время этот текст можно редактировать средствами, аналогичными имеющимся в текстовом процессоре.

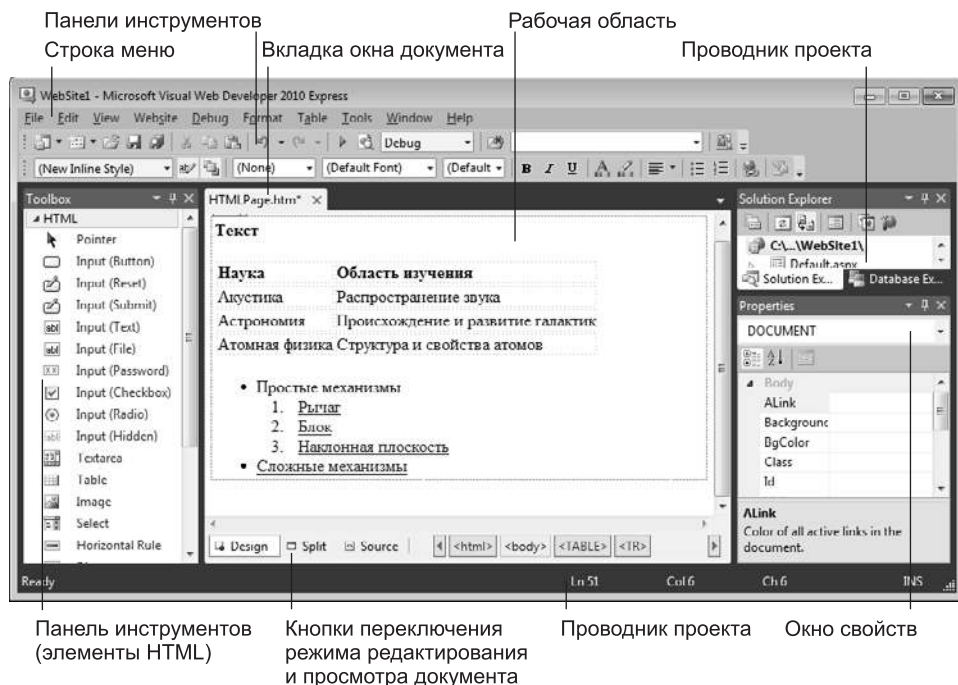


Рис. 19.8. Окно программы Microsoft Visual Web Developer в режиме редактирования веб-страницы

В качестве основного средства форматирования используется панель инструментов **Форматирование** (Formatting), почти идентичная подобной панели текстовых редакторов. Она содержит инструменты для форматирования шрифтов, абзацев, создания списков, гиперссылок.

Для создания гиперссылки надо выделить фрагмент текста, который будет использоваться как ссылка, и нажать на кнопку **Convert to hyperlink** (Конвертировать в гиперссылку) на панели **Форматирование**. При создании ссылки в окне **Hyperlink** (Гиперссылка) следует выбрать тип гиперссылки и ввести саму ссылку в соответствующее поле окна.

Созданная гиперссылка отображается в окне программы так же, как и в окне браузера: синим цветом и с подчеркиванием. Чтобы проверить работоспособность ссылки, следует открыть страницу для просмотра в браузере (**File** ► **View in browser** (Файл ► **Просмотреть в браузере**)) и опробовать ссылки.

Панель **Инструменты** (Toolbox), в части **HTML**, содержит элементы для быстрой вставки в документ различных HTML-тегов. Например, кнопка **Table** позволяет вставить в документ таблицу, **Image** – изображение и т. д.

Чтобы создать таблицу, можно перетащить элемент **Table** из **Панели инструментов** в окно документа и настроить его параметры либо воспользоваться командой меню **Таблица** ► **Вставить таблицу** (**Table** ► **Insert Table**). В открывшемся диалоговом окне указывают размер таблицы (в ячейках), а также дополнительные параметры, реализуемые как атрибуты соответствующих тегов. Чтобы занести информацию

в таблицу, следует установить курсор в нужную ячейку и начать ввод. При работе с ячейками таблицы можно применять любые команды форматирования.

В языке HTML свойства элемента задаются атрибутами тега. Редактор обеспечивает их задание с помощью специального диалогового окна **Свойства (Properties)**. По умолчанию это окно открыто и отображается в правой нижней части окна. Чтобы свойства элемента стали доступными в этом окне, достаточно установить курсор на этот элемент или выделить его. Кроме того, можно воспользоваться кнопками с наименованиями вложенных тегов, которые расположены в нижней части окна редактирования файла. Щелчки по этим кнопкам открывают свойства соответствующих элементов. Элементы управления в этом окне настройки свойств соответствуют атрибутам открывающего тега для выбранного элемента.

Дополнительные объекты вставляются в редактируемый документ при помощи уже упомянутой **Панели инструментов (Toolbox)**.

Следует обратить внимание на то, что состав **Панели инструментов** зависит от типа файла, редактирование которого производится в программе. Так, при разработке динамической .ASPX-страницы разработчику доступно большое количество дополнительных элементов управления.

Иногда требуется вмешаться в процесс автоматического формирования веб-страницы и внести изменения непосредственно в генерируемый код HTML. Для этого надо щелкнуть на кнопке **Исходный код (Source)** в нижней части окна. При этом отображается не примерный вид документа, а сгенерированный программой код HTML (рис. 19.9). Цветная маркировка позволяет немедленно увидеть ключевые слова тегов и названия атрибутов. Этот код можно редактировать вручную, однако в этом случае ответственность за правильность кода переносится с программы на создателя страницы. Возможно, например, некорректное взаимодействие кода, введенного вручную, и элементов HTML, сгенерированных автоматически. Некоторые «ошибки» могут быть даже исправлены автоматически, причем не всегда с ожидаемым результатом.

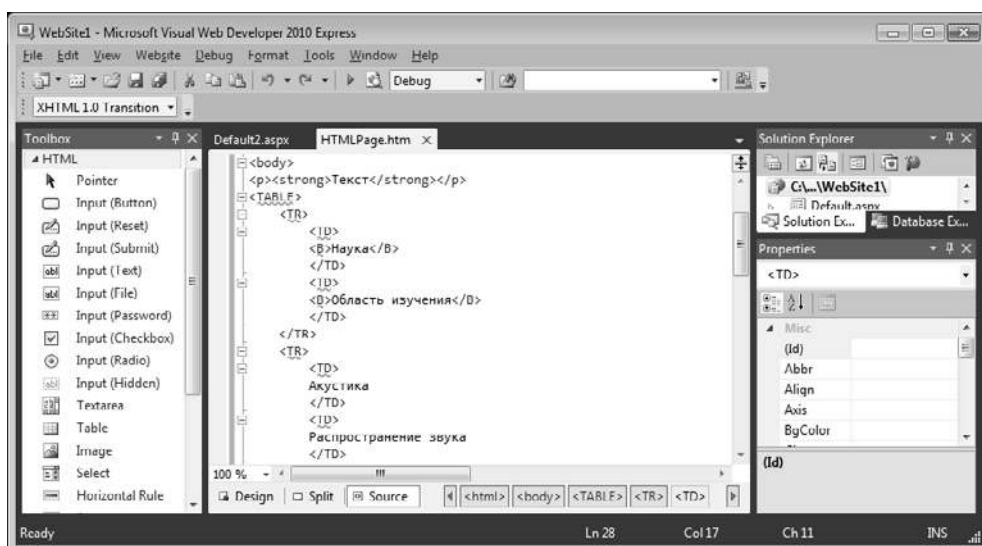


Рис. 19.9. Просмотр автоматически сгенерированного кода HTML веб-страницы

Редактор также позволяет работать в режиме одновременного просмотра кода и внешнего вида веб-страницы. Для включения такого режима нужно воспользоваться кнопкой Split.

Применение шаблонов

Для упрощения и автоматизации создания веб-страниц редактор Microsoft Visual Web Developer позволяет использовать *шаблоны*. Для этого надо создать новый документ командой Файл ► Новый файл (File ► New file) — на экране появится окно для выбора шаблонов, на основе которых можно создавать страницы. Например, это следующие шаблоны:

- ☐ Шаблон Web Form предназначен для создания страниц, которые будут содержать веб-формы.
- ☐ Шаблон HTML Page позволяет создать заготовку для обычной страницы.
- ☐ Шаблон JScript File предназначен для создания JavaScript-файлов. Такие файлы нередко содержат код, который предназначен для оформления интерактивных элементов страниц (таких, как выпадающие меню).
- ☐ Шаблон Style Sheet позволяет создать CSS-файл. Такие файлы обычно используются для придания множеству страниц одинакового оформления.

19.4. Публикация веб-документов

Публикация веб-узла (веб-страниц) состоит в размещении документов HTML и всех сопроводительных файлов (изображений, мультимедиа и пр.) на веб-сервере. Если оставить в стороне организационные вопросы (получение места на веб-сервере, оплата и др.), то остаются две основные проблемы.

- ☐ Как подготовить документы веб-узла таким образом, чтобы перенос их на веб-сервер не привел к нарушению целостности структуры узла?
- ☐ Как произвести копирование файлов на веб-сервер?

Первый вопрос возникает в том случае, если файлы на веб-сервере предполагается разместить в группе тематических каталогов. Он решается с помощью организационных мер в ходе работы над веб-узлом. При формировании будущего узла на своем компьютере необходимо разработать структуру папок и сразу же размещать документы в соответствующих папках. Во внутренних ссылках следует использовать только относительные адреса документов. Перенос файлов и папок на веб-сервер с сохранением структуры сохраняет работоспособность ссылок и корректность подключения вставных объектов (иллюстраций и мультимедиа). При таком подходе облегчается также обновление веб-узла в целом или его отдельных файлов.

Для копирования нужных документов на веб-сервер можно применять как передачу данных на съемном носителе, так и прямое копирование данных через Интернет. Последний способ более надежен. Чтобы воспользоваться им, следует

узнать адреса, используемые при отправке файлов (обычно для публикации веб-страниц применяют протокол FTP).

При использовании специализированных программных средств (например, уже устаревшего, но все еще используемого редактора FrontPage или рассматриваемого нами Microsoft Visual Web Developer) инструменты публикации обычно предоставляет сама программа. Например, в FrontPage для этой цели служит команда **Файл** ► **Опубликовать Web-узел**. В Microsoft Visual Web Developer это команда **Вебсайт** ► **Скопировать вебсайт (Website) ► Copy Web Site**). Во многих случаях организация, предоставляющая место для размещения веб-узла, сама определяет порядок публикации и предоставляет средства для ее выполнения.

Если доступ осуществляется по протоколу *FTP*, необходимо узнать адрес каталога, а также имя пользователя и пароль для доступа к нему (доступ на запись по протоколу *FTP* всегда защищается паролем). Для выполнения переноса файлов желательно использовать специализированный FTP-клиент.

Практическое занятие

Упражнение 19.1. Создание простейшей веб-страницы

1. Запустите текстовый редактор Блокнот (**Пуск** ► **Все программы** ► **Стандартные** ► **Блокнот**).

2. Введите следующий документ:

```
<HTML>
  <HEAD>
    <TITLE>Заголовок документа</TITLE>
  </HEAD>
  <BODY>
    Содержание
    документа
  </BODY>
</HTML>
```

3. Сохраните этот документ под именем `first.htm`.

Перед сохранением в окне сохранения файла выберите в поле **Тип файла** пункт **Все файлы** и введите имя файла вместе с расширением, то есть — `first.htm`

4. Сделайте двойной щелчок по сохраненному файлу. Он откроется в браузере, установленном по умолчанию. При стандартных установках системы это будет Internet Explorer.
5. Посмотрите, как отображается этот файл — простейший корректный документ HTML. Где отображается содержимое элемента `TITLE`? Где отображается содержимое элемента `BODY`?
6. Как отображаются слова «Содержание» и «документа», введенные в двух отдельных строчках? Почему? Проверьте, что происходит при уменьшении ширины окна.

Итог. В этом упражнении мы создали простейший документ HTML. Мы познакомились с особенностями форматирования документов HTML и их отображения при помощи браузера Internet Explorer.

Упражнение 19.2. Изучение приемов форматирования абзацев

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе Блокнот.
2. Удалите весь текст, находящийся между тегами `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега `<BODY>`, а его конкретное содержание может быть любым.
3. Введите заголовок первого уровня, заключив его между тегами `<H1>` и `</H1>`.
4. Введите заголовок второго уровня, заключив его между тегами `<H2>` и `</H2>`.
5. Введите отдельный абзац текста, начав его с тега `<P>`. Пробелы и символы перевода строки можно использовать внутри абзаца произвольно.
6. Введите тег горизонтальной линейки `<HR>`.
7. Введите еще один абзац текста, начав его с тега `<P>`.
8. Сохраните этот документ под именем `paragraph.htm`.
9. Откройте файл двойным щелчком мыши.
10. Посмотрите, как отображается этот файл. Установите соответствие между элементами кода HTML и фрагментами документа, отображаемыми на экране.

Итог. В этом упражнении мы создали документ HTML с разметкой абзацев. Мы определили, как влияют теги HTML на отображение соответствующих частей документа.

Упражнение 19.3. Создание гиперссылок

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе Блокнот.
2. Удалите весь текст, находящийся между тегами `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега `<BODY>`.
3. Введите фразу: Текст до ссылки.
4. Введите тег: `<A HREF="first.htm">`.
5. Введите фразу: Ссылка.
6. Введите закрывающий тег `</A>`.
7. Введите фразу: Текст после ссылки.
8. Сохраните документ под именем `link.htm`.
9. Откройте файл двойным щелчком мыши.
10. Убедитесь в том, что текст между тегами `<A>` и `</A>` выделен как ссылка (цветом и подчеркиванием).

11. Щелкните на ссылке и убедитесь, что при этом загружается документ, на который указывает ссылка.
12. Щелкните на кнопке **Назад** на панели инструментов, чтобы вернуться к предыдущей странице. Убедитесь, что ссылка теперь считается «просмотренной» и отображается другим цветом.

Итог. В этом упражнении мы создали документ HTML, содержащий гиперссылки. Мы увидели, как гиперссылки отображаются в документе, и научились пользоваться ими.

Упражнение 19.4. Создание изображения и использование его на веб-странице

Откройте программу Paint (Пуск ► Все программы ► Стандартные ► Paint). Задайте размеры нового рисунка, например 50 × 50 точек.

1. Выберите красный цвет переднего плана и зеленый цвет фона. Залейте рисунок фоновым цветом.
2. Инструментом Кисть нанесите произвольный красный рисунок на зеленый фон.
3. Сохраните рисунок под именем pic1.gif (в формате GIF).
4. Если это упражнение выполняется не сразу после предыдущего, откройте документ first.htm в программе Блокнот.
5. Удалите весь текст, находящийся между тегами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY>.
6. Введите произвольный текст (протяженностью 4–5 строк) и установите текстовый курсор в его начало.
7. Введите тег .
8. Сохраните документ под именем picture.htm.
9. Откройте документ двойным щелчком мыши.
10. Посмотрите на получившийся документ, обращая особое внимание на изображение.
11. Вернитесь в программу Блокнот и измените значение атрибута: ALIGN="TOP". Сохраните файл под тем же именем.
12. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.
13. Вернитесь в программу Блокнот и измените значение атрибута: ALIGN=»LEFT». Сохраните файл под тем же именем.
14. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.

15. Вернитесь в программу Блокнот и добавьте в тег `<IMG>` атрибуты: `HSPACE=40` `VSPACE=20`. Сохраните файл под тем же именем.
16. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.

Итог. В этом упражнении мы научились вставлять изображения в документ. Мы выяснили, как влияют атрибуты тега `<IMG>` на способ отображения изображения.

Упражнение 19.5. Приемы форматирования текста

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе Блокнот.
2. Удалите весь текст, находящийся между тегами `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега `<BODY>`, а его конкретное содержание может быть любым.
3. Введите тег `<BASEFONT SIZE="5" COLOR="BROWN">`. Он задает вывод текста по умолчанию увеличенным шрифтом и коричневым цветом.
4. Введите произвольный абзац текста, который будет выводиться шрифтом, заданным по умолчанию. Начните этот абзац с тега `<P>`.
5. Введите теги: `<P><FONT SIZE="-2" FACE="ARIAL" COLOR="GREEN">`.
6. Введите очередной абзац текста, закончив его тегом `</FONT>`.
7. В следующем абзаце используйте по своему усмотрению парные теги: `<B>` (полужирный шрифт), `<I>` (курсив), `<U>` (подчеркивание), `<S>` (вычеркивание), `<SUB>` (нижний индекс), `<SUP>` (верхний индекс).
8. В следующем абзаце используйте по своему усмотрению парные теги: `<EM>` (выделение), `<STRONG>` (сильное выделение), `<CODE>` (текст программы), `<KBD>` (клавиатурный ввод), `<SAMP>` (пример вывода), `<VAR>` (компьютерная переменная).
9. Сохраните полученный документ под именем `format.htm`.
10. Откройте файл `format.htm` двойным щелчком мыши.
11. Изучите, как использованные элементы HTML влияют на способ отображения текста.
12. Вернитесь в программу Блокнот и измените документ так, чтобы элементы, задающие форматирование, были вложены друг в друга. Сохраните документ под тем же именем.
13. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы.

Итог. Мы познакомились с некоторыми элементами языка HTML, которые могут использоваться для форматирования текста документа. Мы выяснили, как эти элементы воздействуют на отображение документа, и узнали, что такие элементы можно вкладывать друг в друга.

Упражнение 19.6. Приемы создания списков

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе Блокнот.
2. Удалите весь текст, находящийся между тегами `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега `<BODY>`, а его конкретное содержание может быть любым.
3. Вставьте в документ тег `<OL TYPE="I">`, который начинает упорядоченный (нумерованный) список.
4. Вставьте в документ элементы списка, предваряя каждый из них тегом `<LI>`.
5. Завершите список при помощи тега `</OL>`.
6. Сохраните полученный документ под именем `list.htm`.
7. Откройте файл `list.htm` двойным щелчком мыши.
8. Изучите, как упорядоченный список отображается в программе Internet Explorer, обращая особое внимание на способ нумерации, заданный при помощи атрибута `TYPE=`.
9. Вернитесь в программу Блокнот и установите текстовый курсор после окончания введенного списка.
10. Вставьте в документ тег `<UL TYPE="SQUARE">`, который начинает неупорядоченный (маркированный) список.
11. Вставьте в документ элементы списка, предваряя каждый из них тегом `<LI>`.
12. Завершите список при помощи тега `</UL>`. Сохраните документ под тем же именем.
13. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы, обратив внимание на способ маркировки, заданный при помощи атрибута `TYPE=`.
14. Вернитесь в программу Блокнот и установите текстовый курсор после окончания введенного списка.
15. Вставьте в документ тег `<DL>`, который начинает список определений.
16. Вставьте в список определяемые слова, предваряя соответствующие абзацы тегом `<DT>`.
17. Вставьте в список соответствующие определения, предваряя их тегом `<DD>`.
18. Завершите список при помощи тега `</DL>`. Сохраните документ под тем же именем.
19. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как выглядит при отображении веб-страницы список определений.

Итог. Мы научились создавать списки средствами языка HTML и определять способ их нумерации (маркировки). Мы установили, как выглядят списки при их отображении в браузере Internet Explorer. Мы также научились создавать список определений.

Упражнение 19.7. Создание таблиц

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе Блокнот.
2. Удалите весь текст, находящийся между тегами `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега `<BODY>`. В данном упражнении используется список номеров телефонов.
3. Введите тег `<TABLE BORDER="10" WIDTH="100%">`.
4. Введите строку: `<CAPTION ALIGN="TOP">Список телефонов</CAPTION>`.
5. Первая строка таблицы должна содержать заголовки столбцов. Определите ее следующим образом:

```
<TR BGCOLOR="YELLOW" ALIGN="CENTER">  
<TH>Фамилия<TH>Номер телефона
```
6. Определите последующие строки таблицы, предваряя каждую из них тегом `<TR>` и помещая содержимое каждой ячейки после тега `<TD>`.
7. Последнюю строку таблицы задайте следующим образом:

```
<TR><TD ALIGN="CENTER" COLSPAN="2">На первом этаже здания имеется бесплатный телефон-автомат.
```
8. Завершите таблицу тегом `</TABLE>`.
9. Сохраните документ под именем `table.htm`.
10. Двойным щелчком мыши откройте файл `table.htm`.
11. Изучите, как созданная таблица отображается в программе Internet Explorer, обращая особое внимание на влияние заданных атрибутов.
12. Измените ширину окна браузера и установите, как при этом изменяется внешний вид таблицы.

Итог. Мы познакомились с приемами создания таблиц средствами языка HTML для представления данных. Мы научились создавать таблицы и изменять их вид при помощи атрибутов тегов HTML.

Упражнение 19.8. Создание описания фреймов

1. Запустите текстовый редактор Блокнот (Пуск ► Все программы ► Стандартные ► Блокнот).
2. Введите следующий документ:

```
<HTML>  
<HEAD>  
  <TITLE>Описание фреймов</TITLE>  
</HEAD>  
<FRAMESET ROWS="60%,*">  
  <FRAME SRC="table.htm">  
  <FRAMESET COLS="35%.65%" NORESIZE>  
    <FRAME SRC="first.htm">  
    <FRAME SRC="links.htm">  
  </FRAMESET>  
</FRAMESET>  
</HTML>
```

3. Сохраните этот документ под именем `frames.htm`.
4. Двойным щелчком мыши откройте файл `frames.htm`.
5. Изучите представление нескольких созданных ранее документов в отдельных фреймах.
6. Посмотрите, что происходит при изменении ширины окна браузера.
7. Проверьте, можно ли изменить положение границ фреймов методом перетаскивания при помощи мыши.
8. Щелкните на ссылке, имеющейся в одном из фреймов, и посмотрите, как будет отображен новый документ.
9. Щелкните на кнопке **Назад** на панели инструментов и убедитесь, что возврат к предыдущему документу не нарушает структуру фреймов.
10. Вернитесь в программу Блокнот и измените структуру и параметры фреймов по своему усмотрению. Сохраните документ под тем же именем.
11. Вернитесь в программу Internet Explorer и щелкните на кнопке **Обновить** на панели инструментов. Убедитесь, что измененный вид веб-страницы соответствует замыслу. Если это не так, вернитесь в программу Блокнот, найдите и исправьте ошибки.

Итог. Мы научились отображать в рамках одной веб-страницы несколько документов при помощи фреймов. Мы узнали, как фреймы отображаются в программе Internet Explorer. Мы исследовали особенности навигации с помощью гиперссылок по веб-странице, содержащей фреймы.

Упражнение 19.9. Создание веб-документа с помощью редактора Microsoft Visual Web Developer

1. Запустите программу Microsoft Visual Web Developer (Пуск ► Все программы ► Microsoft Visual Studio 2010 ► Microsoft Visual Web Developer).
2. Выполните команду Файл ► Новый файл (File ► New file), в появившемся окне выбора шаблона выберите HTML Page.
3. Переключитесь в режим визуального проектирования страницы (кнопка Design).
4. Введите в автоматически созданную веб-страницу произвольный текст документа.
5. С помощью панели инструментов Форматирование (Formatting) отформатируйте текст по собственному усмотрению.
6. Для создания таблицы выполните команду Таблица ► Добавить таблицу (Table ► Insert Table).
7. Для добавления иллюстраций используйте элемент Image с панели инструментов. После того как заготовка рисунка добавлена, настройте его свойство, используя окно Свойства (Properties). Если окно на экране не отображено, вызовите его соответствующей командой контекстного меню элемента. Иллюстрации возьмите из папки \Windows.

8. Дайте команду Файл ► Сохранить (File ► Save) и задайте имя файла wysiwyg.htm.
9. Дайте команду Файл ► Просмотр в браузере (File ► View in browser).
10. Убедитесь, что созданный документ правильно отображается браузером. Обратите внимание на наличие отличий вида документа при отображении в браузере и в программе Microsoft Visual Web Developer.
11. Измените ширину окна браузера и посмотрите, как при этом меняется вид документа.
12. Вернитесь в программу Microsoft Visual Web Developer и щелкните на кнопке просмотра исходного кода (Source).
13. Изучите автоматически сгенерированный код HTML, определите, как с помощью тегов HTML реализованы использованные команды форматирования.
14. Оцените качество получившегося кода HTML.

Итог. Мы научились использовать редактор Microsoft Visual Web Developer для создания веб-документов. Мы узнали, как с его помощью форматируют текст документа, создают таблицы и вставляют изображения. Мы исследовали полученный при этом код HTML.

Исследовательская работа

Задание. Исследование методов создания абзацного отступа в документах HTML

Язык HTML не содержит «официальных» средств для создания абзацных отступов. Все браузеры, предназначенные для вывода текста на экран компьютера, выводят текст без отступа, вставляя пустую строку между отдельными абзацами. Создание абзацного отступа, таким образом, требует использования специальных приемов.

1. Запустите текстовый редактор (например, Блокнот) и начните создание документа HTML. Введите теги структурных элементов и дайте документу заголовки, например Имитация абзацных отступов.
2. Введите небольшой абзац текста, который будет использоваться как эталон. Сохраните документ.
3. Запустите браузер Internet Explorer и откройте в нем созданный документ. Отрегулируйте ширину окна программы так, чтобы исследуемый абзац занимал несколько строк. Убедитесь, что он выводится без отступа.
4. *Добавление пробелов.* Вернитесь к редактированию документа. Разместите после эталонного абзаца горизонтальную линейку (тег <HR>). Скопируйте эталонный абзац через буфер обмена, поместив копию ниже линейки. Добавьте в начало скопированного абзаца несколько пробелов. Сохраните документ.
5. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить. Посмотрите на добавленный абзац. Объясните, почему создать абзацный отступ таким образом не удастся.

6. *Метод элемента списка.* Вернитесь к редактированию документа. Разместите после последнего абзаца горизонтальную линейку. Скопируйте эталонный абзац через буфер обмена, поместив копию ниже линейки. Добавьте в начало скопированного абзаца тег `<DD>`. Сохраните документ.
7. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить. Посмотрите на добавленный абзац. Убедитесь, что появился абзацный отступ. Правильно ли используется код HTML в полученном документе? Можно ли рекомендовать такой метод создания абзацного отступа? Почему?
8. *Метод неразрывных пробелов.* Вернитесь к редактированию документа. Разместите после последнего абзаца горизонтальную линейку. Скопируйте эталонный абзац через буфер обмена, поместив копию ниже линейки. Добавьте в начало скопированного абзаца повторенную несколько раз комбинацию символов ` `. Сохраните документ.

Комбинации символов, начинающиеся со знака «&» (амперсанд) и заканчивающиеся точкой с запятой, служат в языке HTML для задания символов, которые отсутствуют на клавиатуре или не могут включаться в текст документа согласно спецификации языка HTML (например, «<»). Комбинация ` ` задает неразрывный пробел.
9. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить. Посмотрите на добавленный абзац. Убедитесь, что появился абзацный отступ. Можно ли рекомендовать такой метод создания абзацного отступа? Почему?
10. *Метод предварительно отформатированного текста.* Вернитесь к редактированию документа. Разместите после последнего абзаца горизонтальную линейку. Скопируйте эталонный абзац через буфер обмена, поместив копию ниже линейки. Добавьте в начало скопированного абзаца тег `<PRE>`, задающий предварительно отформатированный текст, и несколько пробелов. В конце абзаца добавьте тег `</PRE>`. Сохраните документ.
11. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить. Посмотрите на добавленный абзац. Убедитесь, что появился абзацный отступ. Обратите внимание на способ обработки пробелов и символов конца строки в предварительно отформатированном тексте. В чем особенность используемого шрифта? Выскажите свое мнение об использовании этого метода создания абзацного отступа.

Метод невидимого изображения. Вернитесь к редактированию документа. Разместите после последнего абзаца горизонтальную линейку. Скопируйте эталонный абзац через буфер обмена, поместив копию ниже линейки. Добавьте в начало скопированного абзаца тег `<IMG>`, укажите используемый файл изображения и задайте отступ по горизонтали в 10 пикселей (`HSPACE="10"`). Файл изображения должен представлять собой картинку в формате GIF, состоящую из одного пиксела (1 × 1), заданного прозрачным цветом. Сохраните документ.
12. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить. Посмотрите на добавленный абзац. Убедитесь, что появился абзацный отступ.

13. Проанализируйте все использованные методы и выскажите свое мнение о принципиальной целесообразности их использования и о том, какой из них наиболее удобен.

Итог. Мы познакомились с ограничениями, имеющимися в языке HTML, и некоторыми приемами для их преодоления. Мы выяснили, что некоторые эффекты можно обеспечить путем некорректного или нестандартного применения элементов HTML. Мы также познакомились с возможностью ввода символов, отсутствующих на клавиатуре, с помощью специальных последовательностей кодов.

Глава двадцатая

Основы программирования

Компьютерные программы создают *программисты* — люди, обученные процессу их составления (*программированию*). Мы знаем, что программа — это логически упорядоченная последовательность команд, необходимых для управления компьютером (выполнения им конкретных операций), поэтому программирование сводится к созданию последовательности команд, необходимой для решения определенной задачи.

20.1. Языки программирования

Машинный код процессора

Процессор компьютера — это большая интегральная микросхема. Все команды и данные он получает в виде электрических сигналов. Фактически процессор можно рассматривать как огромную совокупность достаточно простых электронных элементов — транзисторов. Транзистор имеет три вывода. На два крайних подается напряжение, необходимое для создания в транзисторе электрического тока, а на средний вывод — напряжение, с помощью которого можно управлять внутренним сопротивлением транзистора, а значит, управлять и током, и напряжением на его выводах.

В электронике транзисторы имеют три применения: для создания усилителей, в электронных схемах, обладающих автоколебательными свойствами, и в электронных переключателях. Последний способ и применяется в цифровой вычислительной технике. В процессоре компьютера транзисторы сгруппированы в микроэлементы, называемые *триггерами* и *вентильями*. Триггеры имеют два устойчивых состояния (открыт — закрыт) и переключаются из одного состояния в другое электрическими сигналами. Этим устойчивым состояниям соответствуют математические понятия 0 или 1. Вентили немного сложнее — они могут иметь несколько входов (напряжение на выходе зависит от комбинаций напряжений на входах) и служат для простейших арифметических и логических операций.

Команды, поступающие в процессор по его шинам, на самом деле являются электрическими сигналами, но и их тоже можно представить как совокупности нулей и единиц, то есть числами. Разным командам соответствуют разные числа. Поэтому реально программа, с которой работает процессор, представляет собой последовательность чисел, называемую *машинным кодом*.

Алгоритм и программа

Управлять компьютером нужно по определенному *алгоритму*. Алгоритм — это точно определенное описание способа решения задачи в виде конечной (по времени) последовательности действий. Такое описание еще называется *формальным*. Для представления алгоритма в виде, понятном компьютеру, служат *языки про-*

граммирования. Сначала всегда разрабатывается алгоритм действий, а потом он записывается на одном из таких языков. В итоге получается текст программы — полное, законченное и детальное описание алгоритма на языке программирования. Затем этот текст программы специальными служебными приложениями, которые называются *трансляторами*, либо переводится в машинный код, либо исполняется.

Что такое язык программирования

Самому написать программу в машинном коде весьма сложно, причем эта сложность резко возрастает с увеличением размера программы и трудоемкости решения нужной задачи. Условно можно считать, что машинный код приемлем, если размер программы не превышает нескольких десятков байтов и нет потребности в операциях ручного ввода/вывода данных.

Поэтому сегодня практически все программы создаются с помощью языков программирования. Теоретически программу можно написать и средствами обычного человеческого (естественного) языка — это называется программированием на *метаязыке* (подобный подход обычно используется на этапе составления алгоритма), но автоматически перевести такую программу в машинный код пока невозможно из-за высокой неоднозначности естественного языка.

Языки программирования — искусственные языки. От естественных они отличаются ограниченным числом «слов», значение которых понятно транслятору, и очень строгими правилами записи команд (*операторов*). Совокупность подобных требований образует *синтаксис* языка программирования, а *смысл* каждой команды и других конструкций языка — его *семантику*. Нарушение формы записи программы приводит к тому, что транслятор не может понять назначение оператора и выдает сообщение о синтаксической ошибке, а правильно написанное, но не отвечающее алгоритму использование команд языка приводит к семантическим ошибкам (называемым еще логическими ошибками или ошибками времени выполнения).

Процесс поиска ошибок в программе называется *тестированием*, процесс устранения ошибок — *отладкой*.

Компиляторы и интерпретаторы

С помощью языка программирования создается не готовая программа, а только ее текст, описывающий ранее разработанный алгоритм. Чтобы получить работающую программу, надо этот текст либо автоматически перевести в машинный код (для этого служат программы-*компиляторы*) и затем использовать отдельно от исходного текста, либо сразу выполнять команды языка, указанные в тексте программы (этим занимаются программы-*интерпретаторы*).

Интерпретатор берет очередной оператор языка из текста программы, анализирует его структуру и затем сразу исполняет (обычно после анализа оператор транслируется в некоторое промежуточное представление или даже машинный код для более эффективного дальнейшего исполнения). Только после того, как текущий оператор успешно выполнен, интерпретатор перейдет к следующему. При этом, если один и тот же оператор должен выполняться в программе многократно,

интерпретатор всякий раз будет выполнять его так, как будто встретил впервые. Вследствие этого, программы, в которых требуется осуществить большой объем повторяющихся вычислений, могут работать медленно. Кроме того, для выполнения такой программы на другом компьютере там также должен быть установлен интерпретатор — ведь без него текст программы является просто набором символов.

По-другому можно сказать, что интерпретатор моделирует некую виртуальную вычислительную машину, для которой базовыми инструкциями служат не элементарные команды процессора, а операторы языка программирования.

Компиляторы полностью обрабатывают весь текст программы (он иногда называется *исходный код*). Они просматривают его в поисках синтаксических ошибок (иногда несколько раз), выполняют определенный смысловой анализ и затем автоматически переводят (*транслируют*) на машинный язык — генерируют машинный код. Нередко при этом выполняется *оптимизация* с помощью набора методов, позволяющих повысить быстродействие программы (например, с помощью инструкций, ориентированных на конкретный процессор, путем исключения ненужных команд, промежуточных вычислений и т. д.). В результате законченная программа получается компактной и эффективной, работает в сотни раз быстрее программы, выполняемой с помощью интерпретатора, и может быть перенесена на другие компьютеры с процессором, поддерживающим соответствующий машинный код.

Основной недостаток компиляторов — трудоемкость трансляции языков программирования, ориентированных на обработку данных сложной структуры, часто заранее неизвестной или динамически меняющейся во время работы программы. Тогда в машинный код приходится вставлять множество дополнительных проверок, анализировать наличие ресурсов операционной системы, динамически их захватывать и освобождать, формировать и обрабатывать в памяти компьютера сложные объекты, что на уровне жестко заданных машинных инструкций осуществить довольно трудно, а для ряда задач практически невозможно.

С помощью интерпретатора, наоборот, допустимо в любой момент остановить работу программы, исследовать содержимое памяти, организовать диалог с пользователем, выполнить сколь угодно сложные преобразования данных и при этом постоянно контролировать состояние окружающей программно-аппаратной среды, благодаря чему достигается высокая надежность работы. Интерпретатор при выполнении каждого оператора проверяет множество характеристик операционной системы и при необходимости максимально подробно информирует разработчика о возникающих проблемах. Кроме того, интерпретатор очень удобен для использования в качестве инструмента изучения программирования, так как позволяет понять принципы работы любого отдельного оператора языка.

В реальных системах программирования перемешаны технологии, и компиляции, и интерпретации. В процессе отладки программа может выполняться по шагам, а результирующий код не обязательно будет машинным — он даже может быть исходным кодом, написанным на другом языке программирования (это существенно упрощает процесс трансляции, но требует компилятора для конечного языка), или промежуточным машинно-независимым кодом абстрактного процессора, который в различных компьютерных архитектурах станет выполняться с помощью интерпретатора или компилироваться в соответствующий машинный код.

Уровни языков программирования

Разные типы процессоров имеют разные наборы команд. Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется *языком программирования низкого уровня*. В данном случае «низкий уровень» не значит «плохой». Имеется в виду, что операторы языка близки к машинному коду и ориентированы на конкретные команды процессора.

Языком самого низкого уровня является *язык ассемблера*, который просто представляет каждую команду машинного кода, но не в виде чисел, а с помощью символьных условных обозначений, называемых *мнемониками*. Однозначное преобразование одной машинной инструкции в одну команду ассемблера называется *транслитерацией*. Так как наборы инструкций для каждой модели процессора отличаются, конкретной компьютерной архитектуре соответствует свой язык ассемблера, и написанная на нем программа может быть использована только в этой среде.

С помощью языков низкого уровня создаются очень эффективные и компактные программы, так как разработчик получает доступ ко всем возможностям процессора. С другой стороны, при этом требуется очень хорошо понимать устройство компьютера, затрудняется отладка больших приложений, а результирующая программа не может быть перенесена на компьютер с другим типом процессора. Подобные языки обычно применяют для написания небольших системных приложений, драйверов устройств, модулей стыковки с нестандартным оборудованием, когда важнейшими требованиями становятся компактность, быстродействие и возможность прямого доступа к аппаратным ресурсам. В некоторых областях, например в машинной графике, на языке ассемблера пишутся библиотеки, эффективно реализующие алгоритмы обработки изображений, требующие интенсивных вычислений.

Языки программирования высокого уровня значительно ближе и понятнее человеку, нежели компьютеру. Особенности конкретных компьютерных архитектур в них не учитываются, поэтому создаваемые программы на уровне исходных текстов легко переносимы на другие платформы, для которых создан транслятор этого языка. Разрабатывать программы на языках высокого уровня с помощью понятных и мощных команд значительно проще, а ошибок при создании программ допускается гораздо меньше.

Поколения языков программирования

Языки программирования принято делить на пять *поколений*. В первое поколение входят языки, созданные в начале 50-х годов, когда первые компьютеры только появились на свет. Это был первый язык ассемблера, созданный по принципу «одна инструкция — одна строка».

Расцвет второго поколения языков программирования пришелся на конец 50-х — начало 60-х годов. Тогда был разработан символический ассемблер, в котором появилось понятие переменной. Он стал первым полноценным языком программирования. Благодаря его возникновению заметно возросли скорость разработки и надежность программ.

Появление третьего поколения языков программирования принято относить к 60-м годам. В это время родились универсальные языки высокого уровня, с их помощью удается решать задачи из любых областей. Такие качества новых языков, как относительная простота, независимость от конкретного компьютера и возможность использования мощных синтаксических конструкций, позволили резко повысить производительность труда программистов. Понятная большинству пользователей структура этих языков привлекла к написанию небольших программ (как правило, инженерного или экономического характера) значительное число специалистов из некомпьютерных областей. Подавляющее большинство языков этого поколения успешно применяется и сегодня.

С начала 70-х годов по настоящее время продолжается период языков четвертого поколения. Эти языки предназначены для реализации крупных проектов, повышения их надежности и скорости создания. Они обычно ориентированы на специализированные области применения, где хороших результатов можно добиться, используя не универсальные, а проблемно-ориентированные языки, оперирующие конкретными понятиями узкой предметной области. Как правило, в эти языки встраиваются мощные операторы, позволяющие одной строкой описать такую функциональность, для реализации которой на языках младших поколений потребовались бы тысячи строк исходного кода.

Рождение языков пятого поколения произошло в середине 90-х годов. К ним относятся также системы автоматического создания прикладных программ с помощью визуальных средств разработки, без знания программирования. Главная идея, которая закладывается в эти языки, — возможность автоматического формирования результирующего текста на универсальных языках программирования (который потом требуется откомпилировать). Инструкции же вводятся в компьютер в максимально наглядном виде с помощью методов, наиболее удобных для человека, не знакомого с программированием.

Обзор языков программирования высокого уровня

FORTRAN (Фортран). Это первый компилируемый язык, созданный Джимом Бэкусом в 50-е годы. Программисты, разрабатывавшие программы исключительно на ассемблере, выражали серьезное сомнение в возможности появления высокопроизводительного языка высокого уровня, поэтому основным критерием при разработке компиляторов Фортрана являлась эффективность исполняемого кода. Хотя в Фортране впервые был реализован ряд важнейших понятий программирования, удобство создания программ было принесено в жертву возможности получения эффективного машинного кода. Однако для этого языка было создано огромное количество библиотек, начиная от статистических комплексов и кончая пакетами управления спутниками, поэтому Фортран продолжает активно использоваться во многих организациях. В частности, в сентябре 2010 года был принят очередной стандарт языка – Fortn 2008 (ISO/IEC 1539-1:2010).

COBOL (Кобол). Это компилируемый язык для применения в экономической области и решения бизнес-задач, разработанный в начале 60-х годов. Он отличается большой «многословностью» — его операторы иногда выглядят как обычные

английские фразы. В Коболе были реализованы очень мощные средства работы с большими объемами данных, хранящимися на различных внешних носителях. На этом языке создано очень много приложений, которые активно эксплуатируются и сегодня. Достаточно сказать, что наибольшую зарплату в США получают программисты на Коболе.

Algol (Алгол). Компилируемый язык, созданный в 1960 году. Он был призван заменить Фортран, но из-за более сложной структуры не получил широкого распространения. В 1968 году была создана версия Алгол 68, по своим возможностям и сегодня опережающая многие языки программирования, однако из-за отсутствия достаточно эффективных компьютеров для нее не удалось своевременно создать хорошие компиляторы.

Pascal (Паскаль). Язык Паскаль, созданный в конце 70-х годов основоположником множества идей современного программирования Никлаусом Виртом, во многом напоминает Алгол, но в нем ужесточен ряд требований к структуре программы и имеются возможности, позволяющие успешно применять его при создании крупных проектов.

Basic (Бейсик). Для этого языка имеются и компиляторы, и интерпретаторы, а по популярности он занимает первое место в мире. Он создавался в 60-х годах в качестве учебного языка и очень прост в изучении.

C (Си). Данный язык был создан в лаборатории Bell и первоначально не рассматривался как массовый. Он планировался для замены ассемблера, чтобы иметь возможность создавать столь же эффективные и компактные программы и в то же время не зависеть от конкретного типа процессора.

Си во многом похож на Паскаль и имеет дополнительные средства для прямой работы с памятью (*указатели*). На этом языке в 70-е годы написано множество прикладных и системных программ и ряд известных операционных систем (Unix).

C++ (Си++). Си++ — это объектно-ориентированное расширение языка Си, созданное Бьярном Страуструпом в 1980 году. Множество новых мощных возможностей, позволивших резко повысить производительность программистов, наложилось на унаследованную от языка Си определенную низкоуровневость, в результате чего создание сложных и надежных программ потребовало от разработчиков высокого уровня профессиональной подготовки.

Java (Джава, Ява). Этот язык был создан компанией Sun в начале 90-х годов на основе Си++. Он призван упростить разработку приложений на основе Си++ путем исключения из него всех низкоуровневых возможностей. Но главная особенность этого языка — компиляция не в машинный код, а в платформенно-независимый байт-код (каждая команда занимает один байт). Этот байт-код может выполняться с помощью интерпретатора — виртуальной Java-машины JVM (Java Virtual Machine), версии которой созданы сегодня для любых платформ. Благодаря наличию множества Java-машин программы на Java можно переносить не только на уровне исходных текстов, но и на уровне двоичного байт-кода, поэтому по популярности язык Ява сегодня занимает второе место в мире после Бейсика.

Особое внимание в развитии этого языка уделяется двум направлениям: поддержке всевозможных мобильных устройств и микрокомпьютеров, встраиваемых в бытовую технику (технология Jini), и созданию платформенно-независимых про-

граммных модулей, способных работать на серверах в глобальных и локальных сетях с различными операционными системами (технология Java Beans). Пока основной недостаток этого языка — невысокое быстродействие, так как язык Ява интерпретируемый.

С# (Си Шарп). В конце 90-х годов в компании Microsoft под руководством Андерса Хейльсберга был разработан язык С#. В нем воплотились лучшие идеи Си и Си++, а также достоинства Java. В 2010 году корпорация Microsoft анонсировала С# версии 4.0., предназначенный для .NET Framework 4.0. и Microsoft Visual Studio 2010.

Языки программирования баз данных

Эта группа языков отличается от алгоритмических языков прежде всего решаемыми задачами. База данных — это файл (или группа файлов), представляющий собой упорядоченный набор *записей*, имеющих единообразную структуру и организованных по единому шаблону (как правило, в табличном виде). База данных может состоять из нескольких таблиц. Удобно хранить в базах данных различные сведения из справочников, картотек, журналов бухгалтерского учета и т. д.

При работе с базами данных чаще всего требуется выполнять следующие операции:

- ☐ создание/модификация свойств/удаление таблиц в базе данных;
- ☐ поиск, отбор, сортировка информации по запросам пользователей;
- ☐ добавление новых записей;
- ☐ модификация существующих записей;
- ☐ удаление существующих записей.

Первые базы данных появились очень давно, как только появилась потребность в обработке больших массивов информации и выборки групп записей по определенным признакам. Для этого был создан *структурированный язык запросов SQL* (Structured Query Language). Он основан на мощной математической теории и позволяет выполнять эффективную обработку баз данных, манипулируя не отдельными записями, а *группами* записей.

Для управления большими базами данных и их эффективной обработки разработаны СУБД (Системы Управления Базами Данных). Практически в каждой СУБД помимо поддержки языка SQL имеется также свой уникальный язык, ориентированный на особенности этой СУБД и не переносимый на другие системы. Сегодня в мире насчитывается три ведущих производителя СУБД: Microsoft (SQL Server), IBM (DB2) и Oracle. Их продукты нацелены на поддержку одновременной работы тысяч пользователей в сети, а базы данных могут храниться в распределенном виде на нескольких серверах. В каждой из этих СУБД реализован собственный диалект SQL, ориентированный на особенности конкретного сервера, поэтому SQL-программы, подготовленные для разных СУБД, друг с другом, как правило, несовместимы.

С появлением персональных компьютеров были созданы так называемые настольные СУБД. Родоначальником современных языков программирования баз

данных для ПК принято считать СУБД dBase II, язык которой был интерпретируемым. Затем для него были созданы компиляторы, появились СУБД FoxPro и Clipper, поддерживающие диалекты этого языка. Сегодня самой распространенной настольной СУБД стала система Microsoft Access.

Языки программирования для Интернета

С активным развитием глобальной сети было создано немало реализаций популярных языков программирования, адаптированных специально для Интернета. Все они отличаются характерными особенностями: языки являются интерпретируемыми, интерпретаторы для них распространяются бесплатно, а сами программы — в исходных текстах. Такие языки называют *скрипт-языками*.

HTML. Общеизвестный язык для оформления документов. Он очень прост и содержит элементарные команды форматирования текста, добавления рисунков, задания шрифтов и цветов, организации ссылок и таблиц. Все веб-страницы написаны на языке HTML или используют его расширения.

Perl. В 80-х годах Ларри Уолл разработал язык Perl. Он задумывался как средство эффективной обработки больших текстовых файлов, генерации текстовых отчетов и управления задачами. По мощности Perl значительно превосходит языки типа Си. В него введено много часто используемых функций работы со строками, массивами, всевозможные средства преобразования данных, управления процессами, работы с системной информацией и др.

PHP. Расмус Лердорф, активно использовавший Perl-скрипты, в 1995 году решил улучшить этот язык, упростив его и дополнив встроенными средствами доступа к базам данных. В результате появилась разработка *Personal Contents Page/Forms Interpreter (PHP/FI)*. Уже через пару лет программы на ее основе использовались на 50 тыс. сайтов. В 1997 году ее значительно усовершенствовали Энди Гутманс и Зив Сураски, и под названием PHP 3.0 этот язык быстро завоевал популярность у создателей динамических сайтов во всем мире.

Tcl/Tk. В конце 80-х годов Джон Аустираут придумал популярный скрипт-язык *Tcl* и библиотеку *Tk*. В *Tcl* он попытался воплотить видение идеального скрипт-языка. Язык *Tcl* ориентирован на автоматизацию рутинных процессов и состоит из мощных команд, предназначенных для работы с абстрактными нетипизированными объектами. Он независим от типа системы и при этом позволяет создавать программы с графическим интерфейсом.

VRML. В 1994 году был создан язык VRML для организации виртуальных трехмерных интерфейсов в Интернете. Он позволяет описывать в текстовом виде различные трехмерные сцены, освещение и тени, текстуры (покрытия объектов), создавать свои миры, путешествовать по ним, «облетать» со всех сторон, вращать в любых направлениях, масштабировать, регулировать освещенность и т. д.

XML. В августе 1996 года WWW-консорциум, ответственный за стандарты на интернет-технологии, приступил к подготовке универсального языка разметки структуры документов, базировавшегося на достаточно давно созданной в IBM технологии SGML. Новый язык получил название XML. Сегодня он служит основой множества системных, сетевых и прикладных приложений, позволяя представлять

в прозрачном для пользователей и программ текстовом виде различные аспекты внутренней структуры иерархически организованных документов. В недалеком будущем он может стать заменой HTML.

Языки моделирования

При создании программ и формировании структур баз данных нередко применяются формальные способы их представления — *формальные нотации*, с помощью которых можно визуально представить (изобразить с помощью мыши) таблицы баз данных, поля, объекты программы и взаимосвязи между ними в системе, имеющей специализированный редактор и генератор исходных текстов программ на основе созданной модели. Такие системы называются *CASE-системами*. В них активно применяются нотации *IDEF*, а в последнее время все большую популярность завоевывает язык графического моделирования *UML*.

Прочие языки программирования

PL/I (ПЛ/1). В середине 60-х годов компания IBM решила взять все лучшее из языков Фортран, Кобол и Алгол. В результате в 1964 году на свет появился новый компилируемый язык программирования, который получил название Programming Language One. В этом языке было реализовано множество уникальных решений, полезность которых удастся оценить только спустя десятилетия, в эпоху крупных программных систем. По своим возможностям ПЛ/1 значительно мощнее многих других языков (Си, Паскаля). Например, в ПЛ/1 присутствует уникальная возможность указания точности вычислений — ее нет даже у Си++ и Явы. Этот язык и сегодня продолжает поддерживаться компанией IBM.

Smalltalk (Смолток). Работа над этим языком началась в 1970 году в исследовательской лаборатории корпорации XEROX, а закончилась спустя 10 лет, воплотившись в окончательном варианте интерпретатора SMALLTALK-80. Данный язык оригинален тем, что его синтаксис очень компактен и базируется исключительно на понятии объекта. В этом языке отсутствуют операторы или данные. Все, что входит в Смолток, является объектами, а сами объекты общаются друг с другом исключительно с помощью сообщений (например, появление выражения $I + 1$ вызывает послышку объекту I сообщения «+», то есть «прибавить», с параметром 1, который считается не числом-константой, а тоже объектом). Больше никаких управляющих структур, за исключением «оператора» ветвления (на самом деле функции, принадлежащей стандартному объекту), в языке нет, хотя их можно очень просто смоделировать.

LISP (Лисп). Интерпретируемый язык программирования, созданный в 1960 году Джоном Маккарти. Ориентирован на структуру данных в форме списка и позволяет организовывать эффективную обработку больших объемов текстовой информации.

Prolog (Пролог). Создан в начале 70-х годов Аланом Колмероз. Программа на этом языке, в основу которого положена математическая модель теории исчисления предикатов, строится из последовательности фактов и правил, а затем формулиру-

ется утверждение, которое Пролог будет пытаться доказать с помощью введенных правил. Человек только описывает структуру задачи, а внутренний «мотор» Пролога сам ищет решение с помощью методов поиска и сопоставления.

Ada (Ада). Назван по имени леди Огасты Ады Байрон, дочери английского поэта Байрона и Анабеллы Милбэнк. В 1980 году сотни экспертов Министерства обороны США отобрали из 17 вариантов именно этот язык, разработанный небольшой группой под руководством Жана Ишбиа. Он удовлетворил на то время все требования Пентагона, а к сегодняшнему дню в его развитие вложены десятки миллиардов долларов. Структура самого языка похожа на Паскаль. В нем имеются средства строгого разграничения доступа к различным уровням спецификаций, доведена до предела мощность управляющих конструкций.

Forth (Форт). Результат попытки Чарльза Мура в 70-х годах создать язык, обладающий мощными средствами программирования, который можно было эффективно реализовать на компьютерах с небольшими объемами памяти, а компилятор мог бы выдавать очень быстрый и компактный код, то есть служил заменой ассемблеру. Однако сложности восприятия программного текста, записанного непривычной форме, сильно затрудняли поиск ошибок, и с появлением Си язык Форт оказался забытым.

Вопросы для самоконтроля

1. Что такое язык программирования?
2. В чем различие компиляторов и интерпретаторов?
3. Объясните термины «язык низкого уровня» и «язык высокого уровня».
4. Расскажите о поколениях языков программирования.
5. Какие языки программирования активно используются сегодня?

20.2. Системы программирования

Средства создания программ

В самом общем случае для создания программы на выбранном языке программирования нужно иметь следующие компоненты.

1. *Текстовый редактор.* Так как текст программы записывается с помощью ключевых слов, обычно происходящих от слов английского языка, и набора стандартных символов для записи всевозможных операций, то формировать этот текст можно в любом редакторе, получая в итоге текстовый файл с *исходным текстом* программы. Лучше использовать специализированные редакторы, которые ориентированы на конкретный язык программирования и позволяют в процессе ввода текста выделять ключевые слова и идентификаторы разными цветами и шрифтами. Подобные редакторы созданы для всех популярных языков и дополнительно могут автоматически проверять правильность синтаксиса программы непосредственно во время ее ввода.

2. Исходный текст с помощью *программы-компилятора* переводится в машинный код. Если обнаружены синтаксические ошибки, то результирующий код создан не будет.

На этом этапе уже возможно получение готовой программы, но чаще всего в ней не хватает некоторых компонентов, поэтому компилятор обычно выдает промежуточный *объектный код* (двоичный файл, стандартное расширение .OBJ).

3. Исходный текст большой программы состоит, как правило, из нескольких *модулей* (файлов с исходными текстами), потому что хранить все тексты в одном файле неудобно — в них сложно ориентироваться. Каждый модуль компилируется в отдельный файл с объектным кодом, которые затем надо объединить в одно целое.

Кроме того, к ним надо добавить машинный код подпрограмм, реализующих различные стандартные функции (например, вычисляющих математические функции `sin` или `ln`). Такие функции содержатся в *библиотеках* (файлах со стандартным расширением .LIB), которые поставляются вместе с компилятором. Сгенерированный код модулей и подключенные к нему стандартные функции надо не просто объединить в одно целое, а выполнить такое объединение с учетом требований операционной системы, то есть получить на выходе программу, отвечающую определенному формату.

Объектный код обрабатывается специальной программой — *редактором связей* или *сборщиком*, который выполняет связывание объектных модулей и машинного кода стандартных функций, находя их в библиотеках, и формирует на выходе работоспособное приложение — *исполнимый код* для конкретной платформы.

Если по каким-то причинам один из объектных модулей или нужная библиотека не обнаружены (например, неправильно указан каталог с библиотекой), то сборщик сообщает об ошибке и готовой программы не получается.

4. Исполнимый код — это законченная программа, которую можно запустить на любом компьютере, где установлена операционная система, для которой эта программа создавалась. Как правило, итоговый файл имеет расширение .EXE.

Интегрированные системы программирования

Итак, для создания программы нужны:

- ☐ текстовый редактор;
- ☐ компилятор;
- ☐ редактор связей;
- ☐ библиотеки функций.

Как правило, в стандартную поставку входят как минимум три последних компонента, но хорошая *интегрированная система* включает в себя и специализированный текстовый редактор, причем почти все этапы создания программы в ней автоматизированы: после того как исходный текст введен, его компиляция

и сборка выполняются одним нажатием клавиши. Это очень удобно, так как не требует ручной настройки множества параметров запуска компилятора и редактора связей, указания им нужных файлов вручную и т. д. Процесс компиляции обычно демонстрируется на экране: показывается, сколько строк исходного текста откомпилировано, или выдаются сообщения о найденных ошибках.

В современных интегрированных системах имеется еще один компонент — *отладчик*, который позволяет анализировать работу программы во время ее выполнения. С его помощью можно последовательно выполнять отдельные операторы исходного текста *по шагам*, наблюдая при этом, как меняются значения различных переменных. Без отладчика разработать крупное приложение очень сложно.

Среды быстрого проектирования

В последние несколько лет в программировании (особенно в программировании для операционной системы Windows) наметился так называемый *визуальный подход*. До этого серьезным препятствием для разработки графических приложений была сложность создания различных элементов управления и контроля их работы. Достаточно взглянуть на окно любой Windows-программы. В нем имеется множество стандартных элементов управления (кнопки, пункты меню, списки, переключатели и т. д.). Очень трудоемко вручную описывать процесс создания этих элементов в соответствии с требованиями Windows, на глазок определять координаты, отслеживать их состояние с помощью специальных команд. Например, для простой программы, складывающей два числа, потребуется один оператор (одна строка исходного текста) для выполнения нужного вычисления и сотни строк кода для подготовки приложения к работе в Windows, создания кнопки и пары полей ввода.

Этот процесс автоматизирован в *средах быстрого проектирования* (Rapid Application Development, RAD-среды). Все необходимые элементы оформления и управления создаются и обслуживаются не путем ручного программирования, а с помощью готовых визуальных *компонентов*, которые с помощью мыши «перетаскиваются» в проектируемое окно. Их свойства и поведение затем настраиваются с помощью простых редакторов, визуально показывающих характеристики соответствующих элементов. При этом вспомогательный исходный текст программы, ответственный за создание и работу этих элементов, генерируется RAD-средой автоматически, что позволяет сосредоточиться только на логике решаемой задачи. В результате программирование во многом заменяется на проектирование — подобный подход называется еще *визуальным программированием*.

Компоненты достаточно легко создавать самостоятельно, поэтому в мире сегодня распространяются тысячи бесплатных и платных компонентов для наиболее известных RAD-сред, из них формируются библиотеки компонентов — *объектные репозитории*. Компоненты выступают в роли «строительных кирпичиков», позволяющих собирать готовое приложение с богатыми возможностями, написав всего десяток строк исходного кода, и такой *компонентный подход* к созданию программ считается очень перспективным, потому что без лишних усилий и на законных основаниях допускает *повторное использование* чужого труда.

Архитектура программных систем

В то время как большинство автономных приложений: офисные программы, среды разработки, системы подготовки текстов и изображений — выполняются на одном компьютере, крупные информационные комплексы (например, система автоматизации предприятия) состоят из десятков и сотен отдельных программ, которые взаимодействуют друг с другом по сети, выполняясь на разных компьютерах. В таких случаях говорят, что они работают в различной *программной архитектуре*. Она делится на следующие группы.

Автономные приложения. Работают на одном компьютере.

Приложения в файл-серверной архитектуре. Компьютеры пользователей системы объединены в сеть, при этом на каждом из них (на *клиентском месте*) запущены копии одной и той же программы, которые обращаются за данными к *серверу* — специальному компьютеру, который хранит файлы, одновременно доступные всем пользователям (как правило, это базы данных). Сервер обладает повышенной надежностью, высоким быстродействием, большим объемом памяти, на нем установлена специальная *серверная* версия операционной системы.

При одновременном обращении нескольких программ к одному файлу, например, с целью его обновления, могут возникнуть проблемы, связанные с неоднозначностью определения его содержимого. Поэтому каждое изменение общедоступного файла выделяется в *транзакцию* — элементарную операцию по обработке данных, имеющую фиксированные начало, конец (успешное или неуспешное завершение) и ряд других характеристик.

Особенность этой архитектуры в том, что все вычисления выполняются на клиентских местах, что требует наличия на них достаточно производительных ПК (это так называемые системы с *толстым клиентом* — программой, которая выполняет всю обработку получаемой от сервера информации).

Приложения в клиент-серверной архитектуре. Эта архитектура похожа на предыдущую, только сервер помимо простого обеспечения одновременного доступа к данным способен еще выполнять программы (обычно выполняются СУБД — тогда сервер называется *сервером баз данных*), которые берут на себя определенный объем вычислений (в файл-серверной архитектуре он реализуется полностью на клиентских местах). Благодаря этому удастся повысить общую надежность системы, так как сервер работает значительно более устойчиво, чем ПК, и снять лишнюю нагрузку с клиентских мест, на которых удастся использовать дешевые компьютеры. Запускаемые на них приложения реально осуществляют небольшие объемы вычислений, а иногда занимаются только отображением получаемой от сервера информации, поэтому они называются *тонкими клиентами*.

Приложения в многозвенной архитектуре. Недостаток предыдущей архитектуры в том, что резко возрастает нагрузка на сервер, а если он выходит из строя, то работа всей системы останавливается. Поэтому в некоторых случаях в систему добавляется так называемый *сервер приложений*, на котором выполняется вся вычислительная работа. Другой сервер баз данных обрабатывает запросы пользователей, на третьем может быть установлена специальная программа — *монитор транзакций*, которая оптимизирует обработку транзакций и балансирует нагрузку на серверы.

В большинстве практических случаев все серверы соединены последовательно — *позвенно*, и выход из строя одного звена если и не останавливает всю работу, то по крайней мере резко снижает производительность системы.

Приложения в распределенной архитектуре. Чтобы избежать недостатков рассмотренных архитектур, были придуманы специальные технологии, позволяющие создавать программу в виде набора компонентов, которые можно запускать на любых серверах, связанных в сеть (компоненты как бы распределены по сети). Основное преимущество подобного подхода в том, что при выходе из строя любого компьютера специальные *программы-мониторы*, которые следят за корректностью работы компонентов и позволяют им «переговариваться» между собой, сразу перезапуская временно пропавший компонент на другом компьютере. При этом общая надежность всей системы становится очень высокой, а вычислительная нагрузка распределяется между серверами оптимальным образом.

Доступ к возможностям любого компонента, предназначенного для общения с пользователем, осуществляется с произвольного клиентского места. При этом, так как все вычисления происходят на серверах, появляется возможность создавать *сверхтонкие клиенты* — программы, только отображающие получаемую из сети информацию и требующие минимальных компьютерных ресурсов. Благодаря этому доступ к компонентной системе возможен не только с ПК, но и с небольших мобильных устройств.

Частный случай компонентного подхода — доступ к серверным приложениям из браузеров через Интернет.

Сегодня наиболее популярны три компонентные технологии — *CORBA* консорциума *OMG*, *Java Beans* компании *Sun* и *COM+/.NET* корпорации *Microsoft*.

Основные системы программирования

Из универсальных языков программирования сегодня наиболее популярны следующие:

- ☐ Бейсик (Basic) — для освоения требует начальной подготовки (общеобразовательная школа);
- ☐ Паскаль (Pascal) — требует специальной подготовки (школы с углубленным изучением предмета и общетехнические вузы);
- ☐ Си++ (C++), Ява (Java), Си Шарп (C#) — требуют профессиональной подготовки (специализированные средние и высшие учебные заведения).

Для каждого из этих языков программирования сегодня имеется немало систем программирования, выпускаемых различными фирмами и ориентированных на различные модели ПК и операционные системы. Наиболее популярны следующие визуальные среды быстрого проектирования программ для *Windows*:

- ☐ *Basic: Microsoft Visual Basic*;
- ☐ *Pascal: Borland Delphi*;
- ☐ *C++: Microsoft Visual C++*;

- *Java: Borland JBuilder;*
- *C#: Microsoft Visual Studio, Borland C#Builder.*

Для разработки серверных и распределенных приложений можно использовать систему программирования Microsoft Visual C++, продукты фирмы Borland, практически любые средства программирования на Java.

В дальнейшем будут рассматриваться возможности, характерные для Бейсика, Паскаля и Си++.

Вопросы для самоконтроля

1. Что нужно для создания программы?
2. Что такое среды быстрого проектирования?
3. Объясните понятие «архитектура программной системы».
4. Опишите основные типы программных архитектур.
5. Какая программная архитектура обеспечивает работу Интернета?

20.3. Алгоритмическое (модульное) программирование

Алгоритм — это формальное описание способа решения задачи путем разбиения ее на конечную по времени последовательность действий (элементарных операций). Под словом «формальное» подразумевается, что описание должно быть абсолютно полным и учитывать все возможные ситуации, которые могут встретиться по ходу решения. Под элементарной операцией понимается действие, которое по заранее определенным критериям (например, очевидности) не имеет смысла детализировать.

Основная идея алгоритмического программирования — разбиение программы на последовательность модулей, каждый из которых выполняет одно или несколько действий. Единственное требование к модулю — чтобы его выполнение всегда начиналось с первой команды и всегда заканчивалось на самой последней (то есть чтобы нельзя было попасть на команды модуля извне и передать управление из модуля на другие команды в обход заключительной).

Алгоритм на выбранном языке программирования записывается с помощью команд описания данных, вычисления значений и управления последовательностью выполнения программы.

Переменные и константы

Реальные данные, с которыми работает программа, — это *числа, строки и логические величины* (аналоги 1 и 0, «да» и «нет», «истина» и «ложь»). Эти типы данных называют *базовыми*.

Каждая единица информации хранится в ячейках памяти компьютера, имеющих свой адреса. На практике заранее неизвестно, в каких конкретно ячейках памяти во время работы программы будут записаны ее данные, поэтому в языках програм-

мирования введено понятие переменной, позволяющее отвлечься от конкретных адресов и обращаться к содержимому памяти с помощью *идентификатора* или *имени* — как правило, последовательности, содержащей английские буквы, цифры, символы подчеркивания и начинающейся не с цифры. Например:

```
Hello
_SumOfReal
x1
H8_G7_F6
```

Это имя будет указывать на *значение*, о реальном адресе и способе хранения которого можно забыть. В процессе работы программы содержимое соответствующих ячеек можно менять, обращаясь к переменной по имени. Лучше выбирать такие названия, которые отражают назначение данной переменной.

Кроме имени и значения, переменная обычно имеет *тип*, определяющий, какая информация хранится в данной переменной (число, строка и т. д.). В зависимости от объема памяти, отведенного для хранения значения переменной, оно должно укладываться в *допустимый диапазон*. Например, значение типа «байт» имеет диапазон от 0 до 255.

Переменные с указанием их типа можно вводить в программу с помощью специальных команд *описания* (*объявления, декларации*). Это позволяет компилятору организовать эффективное хранение и обработку данных и повышает ясность исходных тестов. Каждый тип описывается своим ключевым словом. Значения переменных разных типов допускается преобразовывать друг в друга в соответствии с соглашениями языка программирования. Такой процесс называется *приведением типов*.

Переменные могут существовать на всем протяжении работы программы — тогда они называются *статическими*, а могут создаваться и уничтожаться на разных этапах ее функционирования — такие переменные называются *динамическими*. Все остальные данные в программе, значение которых не меняется на протяжении ее работы, называются *константами* или *постоянными*. Константы, как и переменные, обычно имеют тип. Данные можно указывать явно:

```
123
2.87
"это строка"
```

или для удобства обозначать их идентификаторами. Например, число π , равное 3,1416, можно обозначить как *pi* и везде вместо числа применять идентификатор. Только изменять значение *pi* нельзя, так как это не переменная, а константа.

Числовые данные

Числа обычно бывают двух видов: *целые* и *дробные*. Если число отрицательное, перед ним ставится знак «-»; если положительное, то знак «+» можно ставить, а можно и опускать. Вычисления над целыми числами выполняются *точно*, вычисления над дробными числами — *приближенно*. При записи дробных чисел в качестве десятичного разделителя используется точка:

```
1.28
3.333321
```


Очень большие или очень маленькие числа записываются специальным образом. Для них дополнительно указывается *мантисса* — число со знаком, являющееся степенью числа 10. Мантисса записывается справа от числа через букву *e* (или *E*). Пробелы в такой записи не допускаются.

Например, число 100 (единица, умноженная на 10 во второй степени) запишется так:

1e+2

число 0,003 (тройка, умноженная на 10 в минус третьей степени) так:

3e-3

число со 120 нулями — так:

1E+120

Допускается дробная запись числа с мантиссой:

31.4e-1

Тип числа	Бейсик	Паскаль	Си++
Целое	INTEGER	integer	int
Дробное	DOUBLE	real	float

Арифметические операции

Для записи арифметических действий используются арифметические операторы. В некоторых языках программирования они считаются не операторами, а *операциями*, предназначенными для вычисления значения выражения, но не влияющими на другие значения и не сказывающимися на ходе выполнения программы.

К основным арифметическим операциям относятся:

+ (сложение)

– (вычитание)

* (умножение)

/ (деление)

Такая форма записи отвечает общепринятым соглашениям и принята в большинстве языков программирования.

Каждая арифметическая операция имеет свой *приоритет*. Операции с более высоким приоритетом (умножение и деление) будут выполняться раньше, чем операции с более низким приоритетом (сложение и вычитание). Изменить порядок вычисления выражения можно с помощью круглых скобок.

$b*2 + c/3$

$b*(2 + c) - 3$

Скобки допускается вкладывать друг в друга произвольное число раз. При этом использование квадратных или фигурных скобок, как правило, не допускается.

$((y+2)*3 + 1) / 2$

Арифметические выражения

С помощью арифметических операций формируются арифметические выражения, которые состоят из операций и *операндов* (переменных и констант).

Выражение $i1 + 2$ состоит из одной операции «+» и двух операндов — переменной $i1$ и числовой константы 2.

Каждое выражение имеет значение, которое определяется в момент выполнения оператора, содержащего это выражение. Если на момент вычисления выражения $i1+2$ в переменной $i1$ хранится число 3, то значение этого выражения будет равно $5_{(3+2)}$.

Логические выражения

При создании программ не обойтись без *логических выражений*. Они отличаются тем, что результат их вычислений может принимать только одно из двух допустимых значений — `true` (истина, да, включено) и `false` (ложь, нет, выключено). Чаще всего значение `false` ассоциируется с нулем, а значение `true` — с числом 1 или просто ненулевым значением.

При записи логических выражений используются *операции сравнения* и *логические операции*. Операции сравнения сличают значения правого и левого операндов. Результатом сравнения является `true`, если оно удачно, и `false` в противном случае.

В таблице даны примеры записи операций сравнения для разных языков.

Операция	Варианты написания	
	Бейсик, Паскаль	Си++
Равно	=	==
Не равно	<>	!=
Меньше	<	<
Меньше или равно	<=	<=
Больше	>	>
Больше или равно	>=	>=

```
Pi == 3.14
x > 0
a1 <> b1
```

В одном выражении может потребоваться проверка нескольких подобных условий. Например, надо определить, больше ли значение переменной X , чем 0, и меньше ли, чем 10. Условия могут быть связаны с помощью логических операций, наиболее активно используемые из которых — это И и ИЛИ. В компьютерной графике также часто применяется так называемое исключающее ИЛИ и операция отрицания НЕ. Для нее требуется только один операнд, указывающийся справа от знака операции. Эта операция просто меняет значение своего операнда на противоположное.

1 операнд	2 операнд	И	ИЛИ	исключающее ИЛИ	НЕ (только первый операнд)
true	true	true	true	false	false
true	false	false	true	true	false
false	true	false	true	true	true
false	false	false	false	false	true

В следующей таблице приведен синтаксис записи логических операций.

Логическая операция	Бейсик	Паскаль	Си++
И	AND	and	&&
ИЛИ	OR	or	
НЕ	NOT	not	!

Приоритеты всех логических операций ниже, чем приоритеты операций сравнения, поэтому сравнения всегда выполняются первыми. А логические операции вычисляются в следующем порядке: сначала НЕ, потом И, потом ИЛИ. При необходимости этот порядок может быть изменен с помощью скобок.

Примеры логических выражений:

```
x1 >= 1 && x1 <= 10
(R > 3.14) and (R < 3.149)
(Value < 0) dvalue) OR (Value <= 0)
```

Логический тип

Бейсик	Паскаль	Си++
Базового типа нет. Используется числовой тип INTEGER	boolean	bool

Строчные выражения

Строки в языках программирования всегда заключаются в кавычки. В Си++ и Бейсике для этого используются двойные кавычки, в Паскале — одинарные.

```
"это строка Бейсика или Си++"
'это строка Паскаля'
```

Строка может быть *пустой* — не содержать ни одного символа.

Например:

```
''
""
```

Как правило, строки можно сравнивать друг с другом на эквивалентность (равно и не равно). В некоторых языках программирования допускаются также сравнения типа «больше» или «меньше», при этом происходит последовательное сравнение значений символов (каждый символ представляется в компьютере конкретным числом).

Кроме того, часто допускается также *операция сцепления строк*, записываемая с помощью символа «+». Например:

```
"123" + "4567" — получится "1234567"
"абв " + "abc " + " зюя" — получится "абв abc зюя"
```

Тип «строка»

Бейсик	Паскаль	Си++
STRING	string	Базового типа «строка» нет

Указатели

Некоторые языки программирования допускают в явном виде работу с *указателями* — адресами физической памяти. При этом в них имеется специальная *операция получения адреса* конкретной переменной, что позволяет работать с памятью напрямую, примерно так, как это происходит в языках ассемблера. Такая возможность позволяет добиваться высокой эффективности работы программы, но часто приводит к ошибкам, если указатель вдруг получает неверное значение и при его использовании начинает портиться область памяти, предназначенная совсем для других целей.

Сложные данные

Структуры. До сих пор рассматривались базовые типы данных: числа, строки, логические величины — и операции над базовыми данными. Однако для повышения производительности труда программистов и повышения качества их работы необходимо, чтобы язык программирования имел средства, позволяющие описывать данные в виде, максимально приближенном к их реальным аналогам. Например, чтобы организовать обработку данных по студентам, в программе удобно не просто описать десяток различных переменных, а объединить их в *структуру* (или *запись*) «студент», состоящую из *полей* разного типа «имя», «пол», «год рождения», «группа» и т. д.

Современные языки программирования позволяют применять такие *сложные типы данных*, состоящие из базовых и определенных ранее сложных типов. В результате удастся организовывать структуры данных произвольной сложности: списки, деревья и т. п. При этом структура объединяет группу разных данных под одним названием.

Получить доступ к отдельным составляющим (полям) этой структуры можно по их именам. В рассматриваемых языках программирования такой доступ осуществляется указанием имени структуры и имени поля через точку. Если подобным способом происходит обращение к полю, которое само является структурой, то выделение нужного поля продолжается приписыванием справа имени вложенного поля через точку.

Синтаксис описания структуры

Бейсик	Паскаль	Си++
TYPE имя-структуры поле AS тип ... END TYPE	record поле: тип; ... end;	struct имя { тип поле; ... };

Вот примеры описания структур.

Бейсик:

```
TYPE Student  
Name AS STRING
```

```
Sex AS INTEGER
BirthYear AS INTEGER
END TYPE
```

Паскаль:

```
Record
Name: string;
Sex: boolean;
BirthYear: integer;
end;
```

Си++:

```
struct Student
{
bool Sex;
int BirthYear;
};
```

Доступ к содержимому структуры:

```
Student.BirthYear = 1980;
```

Массивы. Доступ к элементам структуры осуществляется по имени ее составляющих. В одних случаях это значительно повышает наглядность исходных текстов и упрощает процесс программирования, но имеется немало ситуаций, когда надо организовать обработку больших объемов данных одного типа, при этом создавать структуры с сотнями и тысячами полей неразумно. Поэтому в дополнение к структурам в языки программирования введено понятие *массива*, сложного типа данных, доступ к элементам которого происходит по их положению, по номеру или индексу. Например, можно описать массив, состоящий из тысячи элементов численного типа, и затем обратиться к десятому или сотому элементу по его номеру.

При описании массива обычно указывается его *размер* (число элементов) или верхняя и нижняя *границы* — диапазон, в рамках которого можно обращаться к элементам массива.

Синтаксис описания массива

Бейсик	DIM имя (число элементов) AS тип
Паскаль	array[нижняя_граница .. верхняя_граница] of тип;
Си++	тип имя[число-элементов];

В Бейсике нижней границей считается 1, в Си++ — 0, в Паскале она указывается явно.

Вот примеры описания массивов.

Бейсик:

```
DIM IntArray(1000) AS INTEGER
```

Паскаль:

```
array[1..1000] of integer
```

Си++:

```
int IntArray[1000];
```

Доступ к элементу массива осуществляется по его номеру. Этот номер указывается в круглых (Бейсик) или квадратных (Паскаль, Си++) скобках сразу за именем массива (такое действие называется *индексированием*):

```
IntArray( 12 )  
IntArray[ i+1 ]
```

Массивы, границы которых явно заданы в команде описания, называются *статическими*. Их размер известен заранее и не меняется на всем протяжении работы программы.

В последних версиях компилируемых языков программирования реализуются так называемые *динамические* массивы, размер которых может меняться во время выполнения программы. В ряде случаев это весьма удобно, так как позволяет экономно расходовать память, захватывая ее по мере необходимости. Недостаток динамических массивов в том, что организовать эффективную работу с ними, используя компиляторы, сложно. Приходится выполнять множество проверок, связанных с расходом памяти компьютера, что понижает общую эффективность приложения. Динамические массивы в Паскале начали поддерживаться совсем недавно, с активным распространением новых мощных ПК, а в интерпретируемых языках типа Бейсика это было сделано довольно давно.

Во многих языках программирования строки рассматриваются как массивы символов. Их допускается индексировать как обычные массивы.

Правила работы со сложными типами

Отличие базовых типов от сложных в том, что в базовых типах нельзя выделить составные части. При этом поле структуры или элемент массива считаются обычными переменными, и их использование в любых операторах ничем не отличается от использования переменных базовых типов.

В развитых языках программирования допускаются массивы, состоящие из структур, и структуры, состоящие из массивов. При этом возможны достаточно сложные формы записи, например:

```
a[0].Items.Strings[4].value
```

Массив **a** состоит из структур, в описании которых есть поле **Items**, являющееся тоже структурой, имеющей поле **Strings**, которое, в свою очередь, представляет собой массив структур, имеющих поле **value**.

Описание переменных

Чтобы переменную можно было использовать в программе, ее надо предварительно описать, указав ее тип. Пока переменная не описана, обращаться к ней нельзя (хотя в некоторых языках, например в Бейсике и Фортране, считается, что все переменные, не объявленные явно, имеют числовой тип). После того как переменная описана, к ней можно обращаться, но она обычно исходно имеет *неопределенное значение*, поэтому ее надо предварительно *инициализировать* — присвоить ей начальное значение.

Синтаксис команд описания данных

Бейсик	Паскаль	Си++
DIM имя AS тип	var имя: тип;	тип имя;

Вот примеры описания переменных.

Бейсик:

```
DIM X AS DOUBLE
```

Паскаль:

```
var x: real;
var Str: record
    P1: integer;
    S: string;
end;
```

Си++:

```
float x;
int a[20];
```

При описании переменных одного типа в Паскале и Си++ их можно указывать через запятую.

Паскаль:

```
var xx, z2: integer;
```

Си++:

```
int xx, yy[10], z2;
```

Новые типы данных

При определении нескольких переменных со сложной структурой удобно описывать каждую переменную, многократно используя одну и ту же запись структуры. Если, например, в нее потребуется внести изменение (добавить новое поле, изменить тип существующего и т. д.), то придется делать это несколько раз, рискуя ошибиться и пропустить одно из описаний, особенно если они сделаны в разных местах программы.

Чтобы избежать этой проблемы и позволить программистам активно применять нужные структуры данных, в современных языках программирования разрешено определять собственные типы данных, которые допускается использовать в командах описания наравне с базовыми типами.

Синтаксис описания нового типа

Бейсик	Паскаль	Си++
Аналогичен описанию структуры, которое уже является описанием нового типа	type имя = описание;	typedef struct имя-структуры { поля-структуры; } имя; Имя структуры надо указывать только из-за требований синтаксиса. Реально оно нигде не применяется

Название нового типа можно использовать во всех последующих командах описания переменных.

Паскаль:

```
type TMyArray = array[0..99] of integer;
type TMyRecord = record
  Item1: integer;
  Item2: string;
end;
var MyArray: TMyArray;
var R: TMyRecord;
```

Си++:

```
typedef struct name1
{
  int i;
  float x;
} TNewStruct;
TNewStruct NewStruct;
```

Разделение операторов

Если записать подряд несколько операторов и не указать, где кончается один и начинается другой, то в процессе компиляции возникнет множество проблем с выделением отдельных операторов. Поэтому операторы в Паскале и Си++ отделяются друг от друга точкой с запятой «;» (каждый оператор в этих языках должен заканчиваться таким символом), а в Бейсике — двоеточием «:» или переходом на новую строку.

Блок операторов

Часто в программе возникает необходимость выполнить группу операторов (например, в зависимости от какого-либо условия). Такая группа объединяется в *блок* с помощью специальных скобок начала и конца блока, называемых *логическими скобками*.

В Бейсике явного понятия «блок операторов» нет, в Паскале для этого используются ключевые слова `begin` и `end`, а в Си++ — фигурные скобки «{» и «}».

Область действия переменных

Команды описания переменных могут встречаться в разных местах программы. При этом считается, что объявленные в них переменные являются локальными и их *область действия* — текущий блок, в котором они описаны. Как только встречается логическая скобка, закрывающая блок (например, «}»), соответствующая переменная перестает существовать, а выделенная для нее память освобождается.

Некоторые переменные описываются вне блоков и доступны из любого места программы.

Оператор присваивания

Оператор присваивания позволяет изменять текущее значение переменной. Синтаксис его очень простой. В левой части оператора присваивания указывается имя переменной, значение которой изменяется, а справа — выражение, значение которого будет записано в переменную. При этом старое значение, хранившееся в ней, безвозвратно пропадет.

Сам оператор присваивания записывается знаком «=» в Бейсике и Си++ и комбинацией двух знаков «:=» в Паскале (пробел между ними не допускается).

Например:

```
Result = 5
```

В переменную `Result` запишется число 5. Знак «=» означает именно присваивание, а не сравнение, которое может использоваться только в логических выражениях.

Другой пример:

```
X = X + 1
```

Сначала вычисляется значение выражения `X+1`, и затем оно заносится в переменную `X`. Допустима и такая запись:

```
X = X = X
```

Прежде всего выполняется сравнение в правой части (`X = X`), его значение всегда будет `true`, и значением переменной `X` соответственно тоже станет `true`. Для повышения наглядности оператора присваивания в Паскале принята специальная форма его записи:

```
X := X = X;
```

Примеры.

Бейсик:

```
a23 = a22(12) + 1; b1 = b1 - 1
```

Паскаль:

```
a := b*2 + c; d := (e[8] - f)*2.2;
```

Си++:

```
x[5] = y/3.33; y = z[0] - 0.001;
```

Комментарии

При составлении программы очень полезно комментировать различные участки кода, чтобы потом, обратившись к ним, сразу понять, что конкретно выполняется в том или ином месте программы. Забыть смысл того, что было сделано совсем недавно, можно очень быстро — за несколько недель, а в больших проектах и за несколько дней.

Эта проблема становится особенно актуальной, когда группой специалистов разрабатывается объемное приложение, и разобраться в сотнях тысяч строк своего и чужого исходного текста очень сложно.

Языки программирования допускают использование *комментариев* — частей исходных текстов, выделяемых с помощью специальных обозначений и пропускаемых компилятором при анализе текста программы.

Комментарии могут начинаться и заканчиваться особыми символами и охватывать несколько строк кода, а могут записываться только в конце строки — при этом считается, что весь остаток строки является комментарием.

Для обозначения комментариев в одном и том же языке программирования могут использоваться разные символы, поэтому возможно возникновение *вложенных комментариев*. Допустимость такого вложения задается, как правило, в настройках компилятора.

Синтаксис комментария

	Бейсик	Паскаль	Си++
Однострочный комментарий	REM или '	//	//
Многострочный комментарий	нет	{ } или (* *)	/* */

```
X = 5 "комментарий до конца строки
X := 5; // комментарий до конца строки
/*
это комментарий
языка Си++
*/
{
это комментарий
языка Паскаль
(* а это вложенный комментарий *)
}
```

Условный оператор (условные вычисления)

С помощью одного оператора присваивания можно создавать достаточно сложные расчетные программы, однако реализовать абсолютное большинство алгоритмов, просто последовательно выполняя операторы присваивания, невозможно. Постоянно приходится изменять порядок выполнения последовательности вычислений в зависимости от определенных *условий*. Эти условия записываются в виде логических выражений и всегда принимают одно из двух значений — true или false (истинно или ложно). При этом происходит *разветвление* программы — выполнение в дальнейшем может продолжиться с разных операторов.

Синтаксис условного оператора примерно одинаков во всех языках программирования — он представляет собой конструкцию:

```
если условие истинно
то выполнить оператор-1
иначе выполнить оператор-2
```

После ключевого слова IF (*если*) следует условие, и если оно истинно, то выполняется оператор или блок операторов, следующих за ключевым словом THEN (*то*);

если же оно ложно, то выполняется оператор или блок операторов, следующих за ключевым словом ELSE (*иначе*).

Синтаксис описания нового типа

Бейсик	Паскаль	Си++
Аналогичен описанию структуры, которое уже является описанием нового типа	type имя = описание;	typedef struct имя-структуры { поля-структуры; } имя; Имя структуры надо указывать только из-за требований синтаксиса. Реально оно нигде не применяется

Примеры.

Бейсик:

```
IF A <> 0 THEN
  A = 0
ELSE
  A = -1
END IF
```

Паскаль:

```
if a <> 0 then a := 0
             else a := -1;
```

Си++:

```
if( a <> 0 ) a = 0
else a = -1;
```

Вторую часть условного оператора, выполняющуюся в случае, если условие ложно, всегда можно опускать.

Бейсик:

```
IF x < 0 THEN
  y = x / 2
  x = 1
END IF
```

Паскаль:

```
if x < 0 then
begin
  y := x / 2;
  x := 1;
end
```

Си++:

```
if( x < 0 )
{
  y = x / 2;
  x = 1;
};
```

Повторяющиеся вычисления (операторы цикла)

С помощью условных операторов и операторов присваивания теоретически можно реализовать сколь угодно сложный алгоритм. Однако на практике при необходимости организовать обработку тысяч элементов массива (например, присвоить каждому элементу начальное значение) вручную набирать тысячу операторов присваивания крайне тяжело.

Поэтому в языках программирования имеются средства для организации повторных вычислений, называемые *операторами цикла*. Они бывают двух видов: с фиксированным числом повторений и условные операторы цикла.

Каждый оператор цикла состоит из *заголовка цикла*, определяющего число повторений, и *тела цикла* — повторяемого оператора или блока операторов.

Первый вид оператора цикла

При решении задачи примерно в половине случаев заранее известно, сколько раз понадобится выполнить тело цикла. Так бывает, как правило, при обработке массивов, размер которых всегда или известен заранее, или легко определяется.

Заголовок такого оператора состоит из трех частей — *инициализации переменной-счетчика* или *параметра цикла* (присваивания ей начального значения), *определения конечного значения счетчика*, по достижении которого тело цикла надо выполнить в последний раз, и *приращения счетчика*, определяющего, на сколько будет меняться значение счетчика после каждого выполнения тела цикла.

Синтаксис условного оператора

Бейсик	Паскаль	Си++
IF условие THEN оператор-1 ELSE оператор-2 END IF	if условие then оператор-1 else оператор-2; endif	if(условие) оператор-1 else оператор-2; endif

Примеры инициализации тысячи элементов массива *a*.

Бейсик:

```
FOR I = 1 TO 1000  
    A(I) = 0  
NEXT
```

Паскаль:

```
for i := 1 to 1000 do  
    a[i] := 0;
```

Си++:

```
for( i = 0; i < 1000; i = i + 1 )  
    a[i] = 0;
```

В последнем примере счетчик будет принимать значения от 0 до 999, потому что нумерация элементов массива в Си++ начинается с нуля.

Второй вид оператора цикла

Не менее часто встречаются ситуации, когда число повторений заранее неизвестно — надо выполнять цикл, пока не произойдет некоторое событие (пользователь нажмет на кнопку, точность вычислений уложится в заданный порог и т. д.). В таких ситуациях заголовок цикла упрощается. В нем указывается только условие (логическое выражение) — пока его значение равно true, цикл будет выполняться.

Синтаксис оператора цикла

Бейсик	FOR счетчик = начальное_значение TO конечное_значение STEP приращение тело_цикла группа_операторов NEXT Если приращение не указывать, то считается, что оно равно 1
Паскаль	for счетчик := начальное_значение to конечное_значение do оператор или блок операторов; Приращение всегда равно 1
Си++	for(счетчик = начальное_значение; условие_завершения; счетчик = счетчик + приращение) оператор или блок операторов;

Бейсик:

```
DO WHILE A > B
  A = A - 0.01
LOOP
```

Паскаль:

```
while a > b do
  a := a - 0.01;
```

Си++:

```
while( a > b )
  a = a - 0.01;
```

Зацикливание

При использовании условных операторов цикла программиста подстерегает одна опасность. Как показывает практика, достаточно легко сделать ошибку и неверно задать условие окончания цикла, которое всегда будет истинным, — при этом тело цикла станет выполняться бесконечно. Подобная ситуация называется *зацикливанием*.

Например:

```
a = 0; b = 1;
while( a < b )
  a = a - 0.01;
```

Так как исходное значение переменной **a** меньше, чем значение переменной **b**, и это значение будет только уменьшаться, то подобный цикл никогда не закончится.

В некоторых случаях программисты специально применяют подобный трюк, чтобы организовать бесконечный цикл, в котором будут приниматься и обрабатываться внешние сообщения (события). Тогда использование условного оператора цикла может выглядеть так:

```
while true do
  begin
    // тело цикла
  end;
```

Контроль над выходом из цикла при наступлении определенного события при этом полностью возлагается на программиста.

В Бейсике есть специальная форма оператора цикла, позволяющая явно описывать такие бесконечные циклы:

```
DO
  ' тело цикла
LOOP
```

Исключения

Управление порядком выполнения программы может происходить не только с помощью условных операторов и операторов цикла, но и при возникновении *исключений* — ситуаций в программе или операционной системе, требующих немедленного реагирования. Например, при выполнении оператора присваивания и вычислении выражения произошло деление на ноль. Программа остановилась, так как не знает, что ей делать дальше, — ведь получено ошибочное значение. Чаще всего выполнение программы просто прекращается по ошибке, но современные системы разработки позволяют программисту явно контролировать возникновение самых разных исключений (они еще называются *исключительными ситуациями*, требующими немедленного вмешательства) и указывать, какие операторы следует выполнять при их возникновении.

Параллельные вычисления

Еще одна область программирования, в которой возможно изменение явно указанного порядка выполнения операторов, — это область параллельных вычислений. С появлением недорогих ПК с несколькими процессорами возникла возможность *распараллеливания* программы — одновременного выполнения ее независимых частей на разных процессорах, что теоретически позволяет получить выигрыш в быстродействии, линейно зависящий от числа процессоров. Однако на практике это очень сложная задача, которая требует правильного выделения независимых модулей кода (так называемых *процессов*), выполнение которых не скажется на результатах работы других процессов. Так как момент окончания работы того или иного процесса заранее неизвестен, то в программе надо предусмотреть действия, связанные с синхронизацией обработки получаемых результатов. Их выполнение может потребоваться в самые неожиданные моменты, поэтому изменение линейной последовательности работы операторов неизбежно.

Ввод и вывод

Чтобы получать от человека информацию для обработки и показывать результаты своей работы, программа должна иметь средства для организации *интерактивного* общения с пользователем (общения в реальном масштабе времени — человек щелкнул мышкой на кнопке и сразу получил ответ) и средства для ввода данных из файлов и сохранения данных в файлах. Интерактивное общение реализуется с помощью *RAD*-систем, позволяющих быстро спроектировать пользовательский интерфейс. Ввод и вывод информации осуществляется в разных языках по-разному. В Паскале и Бейсике есть операторы для такой работы, в Си++ они выделены в специальные библиотеки. Введен также специальный тип данных «файл» (FILE).

Работа с файлами всегда происходит в три этапа.

1. Файл *открывается* в одном из выбранных режимов (он рассматривается как последовательность строк или двоичных чисел, разрешается только считывать из него данные или только записывать и т. д.). Файл может состоять из последовательности одинаковых блоков, каждый из которых будет представлять собой копию структуры данных определенного типа, описанного в программе. Каждый такой блок называется *записью*.
2. Выполняется считывание, обновление или удаление записей в файле.
3. Файл *закрывается*. Если этого не сделать, то он останется открытым и в дальнейшем к нему нельзя будет обратиться из других программ.

Каждый из этих пунктов реализуется в каждом из языков программирования по-своему. Некоторые пункты требуют для своей реализации нескольких операторов.

Вопросы для самоконтроля

1. Какие типы данных считаются базовыми?
2. Приведите примеры арифметических и логических выражений.
3. Напишите формулу для вычисления среднего арифметического и среднего геометрического значений двух переменных.
4. В чем различие структуры и массива?
5. Зачем нужны комментарии?
6. С помощью условных операторов выполните проверку неравенства $x < y < z$.
7. Из каких частей состоит оператор цикла?
8. Назовите достоинства и недостатки параллельных вычислений.
9. Как организуется работа с файлами?

20.4. Структурное программирование

Подпрограммы

В предыдущем разделе рассматривались основные операторы и типы данных, необходимые для составления программ. При этом предполагалось, что текст программы представляет собой линейную последовательность операторов присваивания, цикла и условных операторов. Таким способом можно решать не очень сложные задачи и составлять программы, содержащие несколько сот строк кода. После этого понятность исходного текста резко падает из-за того, что общая структура алгоритма теряется за конкретными операторами языка, выполняющими слишком детальные, элементарные действия. Возникают многочисленные вложенные условные операторы и операторы циклов, логика становится совсем запутанной, при попытке исправить один ошибочный оператор вносится несколько новых ошибок, связанных с особенностями работы этого оператора, результаты выполнения которого нередко учитываются в самых разных местах программы. Поэтому набрать и отладить длинную линейную последовательность операторов практически невозможно.

При создании средних по размеру приложений (несколько тысяч строк исходного кода) используется *структурное программирование*, идея которого заключается в том, что структура программы должна отражать структуру решаемой задачи, чтобы алгоритм решения был ясно виден из исходного текста. Для этого надо иметь средства для создания программы не только с помощью трех простых операторов, но и с помощью средств, более точно отражающих конкретную структуру алгоритма. С этой целью в программирование введено понятие *подпрограммы* — набора операторов, выполняющих нужное действие и не зависящих от других частей исходного кода. Программа разбивается на множество мелких подпрограмм (занимающих до 50 операторов — критический порог для быстрого понимания цели подпрограммы), каждая из которых выполняет одно из действий, предусмотренных исходным заданием. Комбинируя эти подпрограммы, удастся формировать итоговый алгоритм уже не из простых операторов, а из законченных блоков кода, имеющих определенную смысловую нагрузку, причем обращаться к таким блокам можно по названиям. Получается, что подпрограммы — это новые операторы или операции языка, определяемые программистом.

Возможность применения подпрограмм относит язык программирования к классу *процедурных* языков.

Нисходящее проектирование

Наличие подпрограмм позволяет вести проектирование и разработку приложения *сверху вниз* — такой подход называется *нисходящим проектированием*. Сначала выделяется несколько подпрограмм, решающих самые глобальные задачи (например, инициализация данных, главная часть и завершение), потом каждый из этих модулей детализируется на более низком уровне, разбиваясь, в свою очередь, на небольшое число других подпрограмм, и так происходит до тех пор, пока вся задача не окажется реализованной.

Такой подход удобен тем, что позволяет человеку постоянно мыслить на предметном уровне, не опускаясь до конкретных операторов и переменных. Кроме того, появляется возможность некоторые подпрограммы не реализовывать сразу, а временно откладывать, пока не будут закончены другие части. Например, если имеется необходимость вычисления сложной математической функции, то выделяется отдельная подпрограмма такого вычисления, но реализуется она временно одним оператором, который просто присваивает заранее выбранное значение (например, 5). Когда все приложение будет написано и отлажено, тогда можно приступить к реализации этой функции.

Немаловажно, что небольшие подпрограммы значительно проще отлаживать, что существенно повышает общую надежность всей программы.

Очень важная характеристика подпрограмм — это возможность их *повторного использования*. С интегрированными системами программирования поставляются большие библиотеки стандартных подпрограмм, которые позволяют значительно повысить производительность труда за счет использования чужой работы по созданию часто применяемых подпрограмм.

Рассмотрим пример, демонстрирующий методику нисходящего проектирования. Имеется массив *Осенки*, состоящий из N ($N > 2$) судейских оценок (каждая оценка положительна). В некоторых видах спорта принято отбрасывать самую большую и самую маленькую оценки, чтобы избежать влияния необъективного судейства, а в зачет спортсмену идет среднее арифметическое из оставшихся оценок. Решим эту задачу, постепенно детализируя алгоритм (без привязки к конкретному языку программирования).

Процесс решения наиболее просто описывается подпрограммами:

```
Ввести_оценки_в_массив;  
Удалить_самую_большую_оценку;  
Удалить_самую_маленькую_оценку;  
Рассчитать_среднее_арифметическое_оставшихся_оценок;  
Вывести_результаты;
```

Теперь можно приступить к детализации каждой из этих подпрограмм.

1. Удалить_самую_большую_оценку;

Как удалить самую большую оценку из статического массива? Вместо нее можно просто записать значение 0, а при подсчете среднего арифметического нулевые значения не учитывать.

```
I = Номер_самого_большого_элемента_в_массиве;  
Осенки[ I ] = 0;
```

2. Удалить_самую_маленькую_оценку;

```
I = Номер_самого_маленького_элемента_в_массиве;  
Осенки( I ) = 0;
```

При реализации подпрограммы *Номер_самого_маленького_элемента_в_массиве* надо учесть, что искать придется самое маленькое из *положительных* значений (больших нуля).

3. Рассчитать_среднее_арифметическое_оставшихся_оценок;

Здесь потребуется оператор цикла, вычисляющий сумму всех элементов массива *Осенки*.

```
SUM = 0
FOR I = 1 TO N
    SUM = SUM + Ocenki( I )
NEXT
SUM = SUM / (N - 2)
```

В последнем операторе происходит вычисление среднего арифметического всех оценок. Сумма элементов массива делится на число элементов, уменьшенное на 2, потому что две оценки, самую большую и самую маленькую, учитывать не надо.

Если бы эта задача решалась последовательно, то уже на этапе удаления оценок могли возникнуть определенные проблемы.

Реализацию подпрограмм *Номер_самого_большого_элемента_в_массиве* и *Номер_самого_маленького_элемента_в_массиве* выполните самостоятельно.

Процедуры и функции

Подпрограммы бывают двух видов — *процедуры* и *функции*. Отличаются они тем, что процедура просто выполняет группу операторов, а функция вдобавок вычисляет некоторое значение и передает его обратно в главную программу (*возвращает значение*). Это значение имеет определенный тип (говорят, что функция *имеет* такой-то тип).

В Си++ понятия «процедура» нет — там имеются только функции, а если никакого значения функция не вычисляет, то считается, что она возвращает значение типа «никакое» (void).

Параметры подпрограмм

Чтобы работа подпрограммы имела смысл, ей надо получить данные из внешней программы, которая эту подпрограмму *вызывает*. Данные передаются подпрограмме в виде *параметров* или *аргументов*, которые обычно описываются в ее заголовке так же, как переменные.

Управление последовательностью вызова подпрограмм

Подпрограммы вызываются, как правило, путем простой записи их названия с нужными параметрами. В Бейсике есть оператор CALL для явного указания того, что происходит вызов подпрограммы.

Подпрограммы активизируются *только* в момент их вызова. Операторы, находящиеся внутри подпрограммы, выполняются, только если эта подпрограмма явно вызвана. Пока выполнение подпрограммы полностью не закончится, оператор главной программы, следующий за командой вызова подпрограммы, выполняться не будет.

Подпрограммы могут быть *вложенными* — допускается вызов подпрограммы не только из главной программы, но и из любых других подпрограмм.

В некоторых языках программирования допускается вызов подпрограммы из себя самой. Такой прием называется *рекурсией* и потенциально опасен тем, что может привести к заикливанию — бесконечному самовывозу.

Структура подпрограммы

Подпрограмма состоит из нескольких частей: заголовка с параметрами, тела подпрограммы (операторов, которые будут выполняться при ее вызове) и завершения подпрограммы.

Локальные переменные, объявленные внутри подпрограммы, имеют область действия только ее тело.

Синтаксис оператора цикла

Бейсик	Паскаль	Си++
DO WHILE условие группа операторов LOOP	while условие do оператор или группа операторов;	while(условие) оператор или группа операторов;

Функции

	Бейсик	Паскаль	Си++
Заголовок функции	FUNCTION имя (список_параметров) Тип возвращаемого значения определяется специальным символом после имени функции	function имя (список_параметров): тип_функции;	тип_функции имя(список_параметров)
Тело	Последовательность операторов	begin последовательность операторов end;	{ последовательность операторов };
Завершение	END FUNCTION	Нет	Нет

Как функция возвращает значение

После того как функция рассчитала нужное значение, ей требуется явно вернуть его в вызывающую программу. Для этого может использоваться специальный оператор (return в Си++) или особая форма оператора присваивания, когда в левой части указывается имя функции, а справа — возвращаемое значение.

Далее приведены примеры функции, вычисляющей значение квадрата аргумента.

Бейсик:

```
FUNCTION SQR% (X AS INTEGER)
SQR% = X*X
END FUNCTION
```

Паскаль:

```
function SQR(X: integer): integer;
begin
SQR := X*X
end;
```

```
Си++:  
int SQR(int x)  
{  
    return x*x;  
};
```

Формальные и фактические параметры

Во время создания подпрограммы заранее не известно, какие конкретно параметры она может и будет получать. Поэтому в качестве переменных, выступающих в роли ее аргументов в заголовке, могут использоваться произвольные допустимые названия, даже совпадающие с уже имеющимися. Компилятор все равно поймет, что это не одно и то же.

Параметры, которые указываются в заголовке подпрограммы, называются *формальными*. Они нужны только для описания тела подпрограммы. А параметры (конкретные значения), которые указываются в момент вызова подпрограммы, называются *фактическими* параметрами. При выполнении операторов подпрограммы формальные параметры как бы временно заменяются на фактические.

Пример.

```
int a, y;  
a = 5;  
y = SQR(a);
```

Программа вызывает функцию SQR() с одним фактическим параметром a. Внутри подпрограммы формальный параметр x получает значение переменной a и возводится в квадрат. Результат возвращается обратно в программу и присваивается переменной y.

Событийно-ориентированное программирование

С активным распространением системы Windows и появлением визуальных RAD-сред широкую популярность приобрел событийный подход к созданию программ — *событийно-ориентированное программирование*.

Идеология системы Windows основана на событиях. Щелкнул человек на кнопке, выбрал пункт меню, нажал на клавишу или кнопку мыши — в Windows генерируется подходящее *сообщение*, которое отсылается окну соответствующей программы.

Структура программы, созданной с помощью событийного программирования, следующая. Главная часть представляет собой один бесконечный цикл, который опрашивает Windows, следя за тем, не появилось ли новое сообщение. При его обнаружении вызывается подпрограмма, ответственная за *обработку* соответствующего события (обрабатываются не все события, их сотни, а только нужные), и подобный цикл опроса продолжается, пока не будет получено сообщение «Завершить работу».

События могут быть *пользовательскими*, возникшими в результате действий пользователя, *системными*, возникающими в операционной системе (например,

сообщения от таймера), и *программными*, генерируемыми самой программой (например, обнаружена ошибка и ее надо обработать).

Событийное программирование является развитием идей нисходящего проектирования, когда постепенно определяются и детализируются реакции программы на различные события.

Вопросы для самоконтроля

1. С какой целью применяют подпрограммы?
2. Чем характеризуются процедурные языки программирования?
3. В чем состоит идея нисходящего проектирования?
4. Что общего и в чем отличия процедуры и функции?
5. Определите значение выражения $F(1,2) + F(10,0.1)$, если функция $F(a,b)$ рассчитывается как $a*a + b*b$.
6. В чем различие между событийным и структурным программированием?
7. Как организуется обработка программных событий?

20.5. Объектно-ориентированное программирование

Понятие объекта

Развитие идей структурного и событийного программирования существенно подняло производительность труда программистов и позволило в разумные сроки (несколько месяцев) создавать приложения объемом в сотни тысяч строк. Однако такой объем уже приблизился к пределу возможностей человека, и потребовались новые технологии разработки программ.

В начале 80-х годов в программировании возникло новое направление, основанное на понятии *объекта*. До того времени основные ограничения на возможность создания больших систем накладывала разобщенность в программе данных и методов их обработки.

Реальные объекты окружающего мира обладают тремя базовыми характеристиками: они имеют набор свойств, способны разными методами изменять эти свойства и реагировать на события, возникающие как в окружающем мире, так и внутри самого объекта. Именно в таком виде в языках программирования и реализовано понятие *объекта* как совокупности *свойств* (структур данных, характерных для этого объекта), *методов* их обработки (подпрограмм изменения свойств) и *событий*, на которые данный объект может реагировать и которые приводят, как правило, к изменению свойств объекта.

Появление возможности создания объектов в программах качественно повлияло на производительность труда программистов. Максимальный объем приложений, которые стали доступны для создания группой программистов из 10 человек, за несколько лет увеличился до миллионов строк кода, при этом одновременно удалось добиться высокой надежности программ и, что немаловажно, повторно использовать ранее созданные объекты в других задачах.

Класс

Объекты могут иметь идентичную структуру и отличаться только значениями свойств. В таких случаях в программе создается новый тип, основанный на единой структуре объекта (по аналогии с тем, как создаются новые типы для структур данных). Он называется *классом*, а каждый конкретный объект, имеющий структуру этого класса, называется *экземпляром класса*.

Описание нового класса

Описание нового класса похоже на описание новой структуры данных, только к полям (свойствам) добавляются методы — подпрограммы.

В Си++ и Паскале для описания класса используется ключевое слово `class`.

Паскаль:

```
class TMyClass
Item1: integer;
Item2: string;
function GetSum(n: integer): integer;
procedure Initialize;
end;
```

Си++:

```
class TMyClass
{
int Item1;
int Item2;
int GetSum(int n);
void Initialize();
};
```

При определении подпрограмм, принадлежащих конкретному классу, его методов в заголовке подпрограммы перед ее названием явно указывается, к какому классу она принадлежит. Название класса от названия метода отделяют специальные символы (точка в Паскале или два двоеточия в Си++).

Паскаль:

```
procedure TMyClass.Initialize;
begin
Item1 := 1;
Item2 := "";
end;
```

Си++:

```
void TMyClass::Initialize()
{
Item1 = 1;
Item2 = 0;
}
```

Класс — это тип данных, такой же, как любой другой базовый или сложный тип. На его основе можно описывать конкретные объекты (экземпляры классов).

Паскаль:

```
var C1, C2: TMyClass;
```

Си++:

```
TMyClass C1, C2;
```

Доступ к свойствам объектов и к их методам осуществляется так же, как к полям записей, через точку:

```
C1.Item1 := 5;  
C2.Initialize;  
x := C1.GetSum(21);
```

Объектно-ориентированное программирование базируется на трех ключевых концепциях — инкапсуляции, наследовании и полиморфизме. Объединение данных с методами в одном типе (классе) называется *инкапсуляцией*. Помимо объединения, инкапсуляция позволяет ограничивать доступ к данным объектов и реализации методов классов. В результате у программистов появляется возможность использования готовых классов в своих приложениях на основе только описаний этих классов.

Наследование

Важнейшая характеристика класса — возможность создания на его основе новых классов с *наследованием* всех его свойств и методов и добавлением собственных. Класс, не имеющий предшественника, называется *базовым*.

Например, класс «животное» имеет свойства «название», «размер», методы «идти» и «размножаться». Созданный на его основе класс «кошка» наследует все эти свойства и методы, к которым дополнительно добавляется свойство «окраска» и метод «пить».

Наследование позволяет создавать новые классы, повторно используя уже готовый исходный код и не тратя времени на его переписывание.

Полиморфизм

В большинстве случаев методы базового класса у классов-наследников приходится переопределять — объект класса «кошка» выполняет метод «идти» совсем не так, как объект класса «амеба». Все переопределяемые методы по написанию (названию) будут совпадать с методами базового объекта, однако компилятор по типу объекта (его классу) распознает, какой конкретно метод надо использовать, и не вызовет для объекта класса «кошка» метод «идти» класса «животное». Такое свойство объектов переопределять методы наследуемого класса и корректно их использовать называется *полиморфизмом*.

Визуальное программирование

Технологии объектного, событийного и структурного программирования сегодня объединены в RAD-системах, которые содержат множество готовых классов, представленных в виде визуальных *компонентов*, которые добавляются в программу одним щелчком мыши. Программисту надо только спроектировать внешний вид окон своего приложения и определить обработку основных событий — какие опе-

раторы будут выполняться при нажатии на кнопки, при выборе пунктов меню или щелчках мышкой. Весь вспомогательный исходный код среда сгенерирует сама, позволяя программисту полностью сосредоточиться только на реализации алгоритма.

Вопросы для самоконтроля

1. Для чего в языки программирования было введено понятие класса?
2. В чем различие между классом и объектом?
3. Поясните понятие инкапсуляции на бытовых примерах.
4. Для чего применяется механизм наследования?
5. Как полиморфизм модифицирует принцип наследования?
6. Опишите использование принципов объектно-ориентированного программирования в средах быстрого проектирования.

20.6. Проектирование программ

Программирование как вид деятельности

Появление первых компьютеров породило программирование как *науку*. Разрабатывались первые математические теории обработки информации, средства доказательства правильности программ, оптимизации кода, создания эффективных компиляторов, формального тестирования и т. д. Затем, с появлением универсальных языков программирования третьего поколения, эти аспекты стали менее актуальными — исследования шли и идут в основном в области автоматической генерации исходных текстов и повышения эффективности компиляторов. Программирование превратилось в *искусство* — миллионы людей, не имевших специального образования, получили возможности применять компьютеры для решения собственных прикладных задач, что потребовало от них мастерства создавать правильно работающие программы. Искусством программирование остается и сегодня для профессиональных разработчиков и любителей, создающих программы в одиночку или в небольших компаниях, где все решает индивидуальное мастерство.

Вместе с тем при росте спроса со стороны государственных и частных организаций на все более и более сложные системы автоматизации предприятий, надежные операционные среды, комплексы глобального телекоммуникационного управления возникла необходимость в постановке процесса разработки программного обеспечения (ПО) на поток, превращения программирования в *ремесло*. Было разработано несколько методологий и стандартов, позволивших эффективно организовывать труд сотен программистов средней квалификации, точно укладываться в отпущенные сроки и средства и не зависеть от настроения нескольких талантливых ведущих специалистов. Отрицательная сторона подобных методологий — отсутствие творческого элемента в работе и своеобразная конвейерная «потогонная» система промышленного производства программ, которая, будучи внедренной в организации, в условиях жесточайшего дефицита программистов во всем мире может только отпугнуть сотрудников.

Процедуры

	Бейсик	Паскаль	Си++
Заголовок процедуры	SUB имя (список_параметров)	procedure имя (список_параметров);	void имя(список_параметров)
Тело	Последовательность операторов	begin последовательность операторов end;	{ последовательность операторов };
Завершение	END SUB	Нет	Нет

Экономические аспекты программирования

Когда на свет появились первые компьютеры, одна минута их работы стоила очень дорого, а задачи решались достаточно простые, поэтому в расходах на подготовку программ труд разработчиков составлял небольшую часть. С появлением ПК и ростом спроса на большие программные системы практически всю расходную часть проекта стала составлять зарплата программистов. Как видно из таблицы, большой процент таких проектов заканчивается неудачно, а расходы на них очень велики, поэтому проблемы создания качественного программного обеспечения точно в срок и в рамках бюджета сегодня самые важные и над созданием эффективных методологий производства ПО трудятся специалисты во всех развитых странах.

Этапы разработки программ

Программы небольшого и среднего размера (несколько тысяч строк) создаются, как правило, в два этапа. Сначала необходимо точно установить, что надо сделать, продумать соответствующий алгоритм, определить структуры данных, объекты и взаимодействие между ними (это этап *системного анализа*), а затем выразить этот алгоритм в виде, понятном машине (этап *кодирования*). Если же разрабатывается крупный проект объемом от десятков тысяч до миллионов строк кода, тогда приходится применять специальные методологии проектирования, охватывающие *период разработки ПО*.

Период разработки ПО

Рассмотрим классический период разработки ПО.

1. Формируются и анализируются *требования* к проекту. Этот этап самый важный, так как неправильное формулирование требований приводит к выполнению ненужной работы, а недооценка сложности вызывает перерасход средств и времени. Сегодня около 60% крупных проектов завершаются неудачей именно из-за ошибок на стадии подготовки требований.

На основе требований по различным методикам определяется примерный объем проекта и его трудоемкость, рассчитываются будущие трудозатраты и определяется его *стоимость*. Так как требования к проекту во время ра-

боты над ним могут уточняться и меняться, а выполнение требований надо отслеживать, применяются специальные программы для управления требованиями.

Часто заказчик не в состоянии точно выразить, чего он хочет, и задача специалистов по системному анализу — помочь ему выразить свои требования в виде, пригодном для формализации. После согласования требований подписывается *контракт на разработку ПО*. В дальнейшем любые отклонения от сформулированных требований к продукту (как со стороны заказчика, так и со стороны исполнителя) рассматриваются как нарушение контракта.

Каждый этап требует от заказчика вложения собственных трудозатрат и привлечения высокопрофессиональных специалистов, поэтому он обязательно должен оплачиваться — бесплатно выполнять сложную работу никто не будет. Однако, хотя первый этап самый важный, заказчик редко понимает эту важность и не готов платить достаточно большие суммы. Примерный объем работ на этом этапе — 5% от объема всего проекта.

2. Начинается предпроектное обследование объекта автоматизации. С помощью CASE-средств составляется формальная модель его работы, модель базы данных, объектов и потоков информации. На этом этапе привлекаются специалисты заказчика и эксперты, хорошо знакомые с предметной областью, для которой составляется задача.

Примерный объем работ второго этапа — 10% от общего.

3. На основе формальной модели составляется подробное *техническое задание* для программистов, *спецификации* отдельных модулей, таблицы баз данных, другая сопроводительная документация. Готовится подробный *календарный план работ*, где указываются все сроки, конкретные исполнители и выполняемые объемы работ (понедельно, ежемесячно). Для составления планов работ имеется немало хороших программ, ориентированных как на небольшие группы программистов, так и на коллективы из сотен сотрудников, выполняющих тысячи различных работ в рамках общего плана.

Примерный объем работ третьего этапа — 10% от общего.

4. Выбирается методология разработки ПО и начинается разработка (кодирование). Крупные компании имеют собственные методологии, ориентированные на конкретные задачи (как правило, это задачи автоматизации предприятий), однако все методологии имеют общие черты и более-менее серьезно различающихся известно около двух десятков.

Мы коснулись, в частности, методологий структурного (нисходящего) и объектного проектирования. Хотя объектный подход, безусловно, один из самых передовых, он подразумевает высокую квалификацию всех исполнителей без исключения и поэтому распространен не очень широко.

Достаточно популярна методология *итерационного проектирования*, ориентированная на использование RAD-средств и систем автоматической генерации исходных текстов на основе созданной формальной модели. Такой подход хорош тем, что позволяет быстро создать первый *работающий* прототип программы, когда еще требования к ней окончательно не определены, а в дальнейшем, на

следующих итерациях (их обычно требуется от двух до пяти), постепенно детализировать и реализовывать конкретные возможности, пропущенные по каким-то причинам на предыдущей итерации. Эта методология немного отличается от нисходящего проектирования тем, что применяется, когда окончательные требования неизвестны и могут меняться, а основные работающие функции нужны заказчику как можно быстрее (заказчик чаще хочет получить приложение, законченное на 80%, сегодня, чем законченное на 100% завтра). При нисходящем проектировании основная структура задачи должна быть определена заранее.

Принятие решения о выборе подходящей методологии — очень ответственный процесс. Человек, принимающий такое решение, должен обладать богатым опытом и знаниями в области создания ПО. Многое зависит от инфраструктуры заказчика — какие у него компьютеры, операционные системы, каковы их ресурсы. В соответствии с этим выбираются и средства разработки. Иногда бывает, что лучше всего подходит, например, итерационная методология, но для операционной системы заказчика нет хорошей RAD-среды и приходится остановиться на менее эффективной методологии.

В процессе разработки необходимо:

- непрерывно поддерживать *обратную связь* с заказчиком, чтобы следить за правильностью реализации требований;
- непрерывно контролировать ход работ в соответствии с планом и при отклонениях от него принимать экстренные меры.

Примерный объем этих работ — 10% от общего объема проекта.

5. Когда программа закончена (готова работоспособная *альфа-версия*), она поступает к *тестерам* компании-исполнителя, которые начинают проверять ее на наличие ошибок и сообщать о найденных ошибках программистам. Анализируется, в частности, устойчивость работы программы при вводе недопустимых или критических значений, при отсутствии информации, при неверных действиях, при сбоях аппаратуры, в стрессовых режимах и т. п. Когда число ошибок, выявляемых за определенный срок (неделя, месяц), снижается ниже экспериментально подобранного уровня (на основе аналогичных проектов), начинается *бета-тестирование* программы у заказчика. К такому тестированию привлекается максимально возможное число сотрудников, и программа уже начинает частично функционировать в рабочем режиме.

Примерный объем этих работ — 10% от общего объема проекта.

6. После того как заказчик удовлетворен качеством продукта, начинается его *внедрение* — подготовка к окончательному запуску в эксплуатацию. Если приложение многопользовательское, нередко требуется сформировать и настроить локальную сеть, установить серверы, инсталлировать вспомогательные программы. Очень много проблем возникает при переходе со старых программ (например, бухгалтерского учета, расчета зарплаты, работы с персоналом), которые создавались разными сотрудниками и покупались в разных фирмах (так называемая *лоскутная автоматизация*), к новой интегрированной системе. Необходимо выверить, ввести или перенести множество жизненно важной

информации: всю бухгалтерскую и финансовую отчетность, данные о хранимом оборудовании и множество других сведений. Этот этап самый трудоемкий и «нудный» и занимает порой до 90% времени всего проекта. Для систем автоматизации больших предприятий он растягивается на годы.

7. После того как новая система готова к работе, сотрудников организации заказчика нужно *обучить* работе с этой системой, потому что книг о ней не написано, да и содержится в такой системе, внедренной на конкретном предприятии, множество нюансов, связанных со спецификой работы.

Примерный объем трудозатрат на обучение — 5% от общего объема проекта.

8. После того как заказчик подписывает *акт приемки*, проект считается завершенным, но связь с исполнителем не теряется. Особенно в первое время у пользователей системы постоянно будет возникать множество вопросов по работе с ней. Неизбежно и возникновение ошибок, которые требуется устранять. Кроме того, исполнитель может выпускать новые версии системы, и старая система потребует *обновления*. Сотрудничество с заказчиком по обслуживанию системы называется *сопровождением*. Оно бесплатно на определенный гарантийный срок (например, год).

Реально объем непосредственного программирования и отладки (тестирования) в цикле разработки невелик. Он составляет 10–20% от общего объема работ.

Контроль качества

Чем крупнее проект, тем больше в нем ошибок. При этом слишком затягивать этап тестирования нельзя — нарушаются сроки, растет недовольство потребителей, и на рынок выпускается «сырая» система с множеством ошибок, которые устраняются уже в процессе эксплуатации выпуском многочисленных «заплаток».

Современные технологии создания надежного ПО предусматривают *непрерывный сквозной контроль качества* разрабатываемого продукта на всех этапах жизненного цикла — от анализа требований до внедрения и сопровождения, а не только на этапе тестирования. Качество каждой работы в плане формализуется числовой величиной с помощью специальных методик, но его можно контролировать, только оптимальным образом организовав работу большой группы аналитиков и программистов. Для этого надо иметь возможность *отслеживать* вносимые в проект *изменения* — изменения требований, формальных моделей, сопроводительной документации, версий исходных текстов, хода выполнения календарного плана по разработке, тестированию, внедрению, сопровождению, а также *контролировать и управлять* всеми этапами периода создания программы — *процессом* разработки ПО. Для этого предназначены *системы конфигурационного управления* — сложные и дорогие (десятки и сотни тысяч долларов) продукты, однако без них крупный проект, скорее всего, обречен на неудачу.

Системы не очень сложного конфигурационного управления, охватывающие контроль версий исходных текстов и ряд других аспектов работы группы программистов, встроены, в частности, в такие системы, как *Delphi 7* и *Visual Studio .NET*.

Стандарты качества ПО

Компания может организовать у себя очень эффективный процесс разработки ПО, однако заказчик вполне обоснованно может ей не поверить. Существует международная система сертификации компаний по стандарту качества *ISO 9000*, которая гарантирует, что данная компания выполняет программные проекты в срок и с высоким качеством. Процесс сертификации сложен и требует подготовительной работы в течение нескольких лет.

В университете Карнеги-Меллона в США несколько лет назад была разработана специальная методология *CMM* (*Capability Maturity Model for Software*), позволяющая сертифицировать компании по одному из 5 уровней «зрелости» процесса разработки ПО. Согласно результатам 20-летних исследований Министерства обороны США оказалось, что главная причина слишком частых неудач при разработке крупных информационных проектов заключается, прежде всего, в неумении менеджеров управлять процессом создания качественного ПО.

В отличие от стандарта *ISO 9000*, который просто подтверждает качественную работу компании на основании достаточно общих критериев, методология *CMM* ориентирована именно на качество управления процессом разработки и имеет множество конкретных рекомендаций и указаний по способам организации всех этапов создания ПО. Сегодня в США невозможно получить крупный государственный или военный заказ на создание программного продукта стоимостью более 2 млн долларов, если компания не сертифицирована как минимум по третьему уровню *CMM*. Сегодня на смену этой модели приходит новая интеграционная концепция *CMMI*, дополняющая процессы разработки не менее важными вопросами организации деятельности персонала, общения с подрядчиками, выбора программного обеспечения и т. д.

Повышение индивидуального мастерства

На основе методологии *CMM* была создана методология *PSP* (*Personal Software Process*), ориентированная на индивидуальных разработчиков. Она позволяет в несколько раз повысить качество создания программ, значительно поднять собственную производительность и научиться предсказывать сроки выполнения работ.

Методология *PSP* состоит из 7 этапов самосовершенствования, и чтобы хорошо ее освоить, надо закончить специальные курсы. Однако даже простое знакомство с ее идеологией позволит любому программисту значительно улучшить свою работу.

Перед началом проекта составляется подробный календарный план работ и производится попытка оценить его объем в строках кода и рабочих днях. Весь процесс работы детально хронометрируется, а найденные ошибки подробно описываются — накапливается статистика. По окончании проекта весь процесс тщательно анализируется и делаются выводы о том, что можно улучшить в своей работе, каких ошибок надо стараться избегать, какова реальная производительность труда и т. п. Сегодня лучшая характеристика программиста — это не просто знание *Си++* или *Delphi*, а способность планировать свой труд, разрабатывать программу точно в срок и без ошибок.

Гибкие методики

В последние годы наряду с ростом интереса к «тяжелым» сертификационным методологиям значительно увеличилась роль и популярность так называемых гибких, проворных (*agile*) методик. Они не требуют значительных усилий по реорганизации компании-разработчика и могут быть внедрены не за годы, а за недели. Наиболее известные среди них — экстремальное программирование, *MDA* (архитектура системы, управляемая моделью), *Scrum* и др.

Гибкие методики не навязывают исполнителю множество формализованных действий по разработке ПО. Они предоставляют участникам проекта большую свободу (но и требуют от них большей ответственности) и акцентируют их внимание на важнейших задачах соблюдения требований заказчика, сроков и качества работы.

Методы маркетинга программного обеспечения

Коммерческое ПО. При создании программного продукта *издатель*, выполнив анализ рынка, заказывает у *исполнителя* разработку такого ПО, которое должно пользоваться на рынке спросом, и выделяет на его создание деньги. По окончании работ издатель получает все *имущественные права* на созданный продукт (право на тиражирование, продажу под собственной торговой маркой, право на получение дохода от программы любым способом). При этом может быть оговорено получение исполнителем некоторого процента (роялти) с каждой проданной копии (как правило, для программ, издающихся сотнями тысяч или миллионами копий, роялти составляет 1–3%) — тогда он получает меньшую сумму на разработку или вообще создает программу за свой счет. Если же отчисления не предусмотрены, то все расходы по подготовке программы издатель берет на себя. Он также вкладывает средства в упаковку, рекламную кампанию, организацию сетей сбыта и т. д. Издатель обеспечивает расходы, связанные с сопровождением продукта и технической поддержкой пользователей.

За исполнителем навечно остаются *авторские права* на программу — право указывать свое имя или логотип своей фирмы на начальной заставке, в документации, на упаковочной коробке.

Крупные компании имеют и подразделения разработки ПО, и отделы, занимающиеся его распространением, что помогает эффективно организовать весь процесс от производства программ до доставки их потребителю.

Условно-бесплатное ПО (*shareware*). В связи с активным развитием Интернета огромное число индивидуальных разработчиков получили возможность распространения своих программ по всему миру. Не имея средств на рекламные кампании, они предоставляют возможность получения *ознакомительных версий* их программ (демонстрационных или имеющих искусственные ограничения) через Интернет. Если человеку эта программа нравится, он оплачивает небольшую сумму и получает полную работоспособную версию.

В Интернете есть немало узлов, которые предлагают бесплатные услуги по размещению таких программ. Отечественным *shareware*-разработчикам можно поре-

комендовать сайт www.swrus.com, на котором можно найти множество материалов и форумов по всем вопросам организации shareware.

Бесплатное ПО (freeware, public domain). Такие программы не имеют никаких ограничений, однако автор может *попросить* заплатить ему некоторую сумму, не настаивая, впрочем, на этом (это метод *freeware*). Некоторые программы авторы называют «общественным достоянием» (*public domain*), ничего взамен не требуют и нередко распространяют такое ПО в исходных текстах.

Как правило, стимулом к созданию бесплатных программ служит стремление повысить собственную квалификацию, установить контакты с коллегами, а в случае удачно созданной программы получить известность.

Вопросы для самоконтроля

1. В чем трудности разработки крупных программных проектов?
2. Опишите организацию работы над сложной программной системой.
3. Какой этап разработки проекта является наиболее ответственным?
4. Какова роль собственно программирования в ходе работы над проектом?
5. Опишите известные вам методы контроля качества программного обеспечения.
6. Каковы основные методы распространения программного обеспечения?

20.7. Пример на Бейсике. Разведение кроликов

В данном и последующих разделах рассматриваются три примера, реализованные с помощью разных систем программирования:

- *QBasic* корпорации Microsoft (интерпретирующая версия Бейсика для операционной системы MS-DOS). Эта среда программирования уже много лет используется в учебных целях. Знание принципов программирования на языке Бейсик позволяет в дальнейшем с успехом осваивать такие современные языки программирования, как Microsoft Visual Basic, который позволяет создавать автономные приложения с использованием Microsoft Visual Studio, а также Microsoft Visual Basic for Applications (VBA). VBA встроен во многие приложения прикладного назначения, например, в продукты Microsoft Office, и позволяет эффективно автоматизировать типичные «офисные» задачи.
- *Embarcadero Delphi XE* (система визуального программирования на Паскале);
- *Microsoft Visual Studio 2010* (система программирования на Си++).

Эти примеры включают в себя описание основных приемов работы с данными системами.

Постановка задачи

Итальянский математик Леонардо Фибоначчи придумал оригинальную числовую последовательность, названную в его честь, которая описывает рост численности поколений кроликов. Считается, что каждый год каждая пара животных приносит

приплод — новую пару (самца и самку), которые, в свою очередь, начинают давать приплод через два года (смертность не учитывается). То есть каждый следующий член последовательности равен сумме двух предыдущих, а классическая последовательность Фибоначчи выглядит так:

1, 1, 2, 3, 5, 8, 13, 21, ...

Надо определить, через сколько лет будет достигнута популяция в N особей.

Запуск QBasic

Интерпретатор QBasic входит в стандартную поставку MS-DOS и расположен обычно в каталоге \DOS. Программа-интерпретатор называется qbasic.exe. Для использования интерпретатора в среде Windows 7 QBasic все еще можно загрузить с сайта Microsoft.

После ее запуска на экране появится приветствие, которое пропускается нажатием на клавишу ENTER, после чего QBasic вызывает встроенную справочную систему на английском языке. Она закрывается нажатием клавиши ESC.

Рабочая область экрана (рис. 20.1) поделена на две части. В нижней части, в окне Immediate (Немедленное выполнение) можно вводить операторы Бейсика и тут же их выполнять.

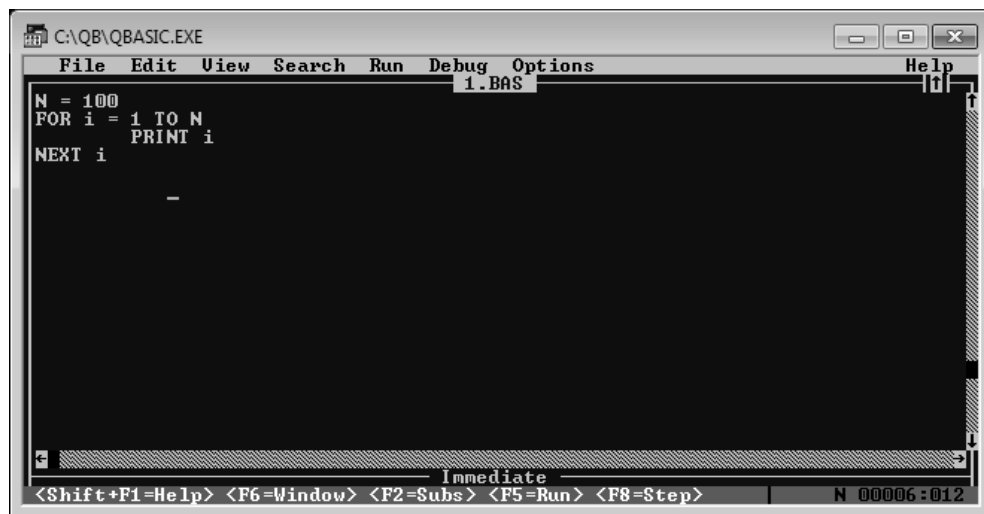


Рис. 20.1. Окно программы QBasic

Вывод на экран

Каждый язык программирования имеет оригинальные средства вывода информации, сильно зависящие от операционной системы. В Бейсике реализован оператор PRINT, который выводит значение следующего за ним выражения на экран, в новую строку.

Для перехода в окно Immediate (Немедленное выполнение) надо нажать клавишу F6. Чтобы сразу получить ответную реакцию от QBasic, достаточно набрать оператор

```
PRINT 2+2
```

и нажать клавишу ENTER, чтобы этот оператор выполнялся.

На экране вывода появится число 4 (результат вычисления выражения 2+2), а в нижней строке — сообщение Press any key to continue (Нажмите любую клавишу для продолжения). Чтобы вернуться в QBasic, надо это сделать.

В операторе вывода можно указывать несколько значений через запятую, тогда они будут выведены в одной строке. Ранее введенный оператор можно изменить, добавив к нему текстовую подсказку:

```
PRINT "Сумма = "; 2+3
```

Если теперь нажать клавишу ENTER, то на экране вывода в новой строке (под ранее напечатанной четверкой) появится фраза

```
Сумма = 5
```

Редактор программы

Таким образом можно познакомиться с работой разных операторов Бейсика, однако выполнять их удастся только поодиночке. Чтобы выполнить группу операторов, их надо объединить в программу.

Набор и редактирование исходного текста программы осуществляется в верхнем окне интерпретатора. Для перехода в него используется клавиша F6.

Решать данную задачу удобнее всего с помощью нисходящего метода. В программе будет бесконечный главный цикл, в котором осуществляется ввод очередного значения количества особей, происходит расчет числа лет, необходимых для размножения, полученный результат печатается, и цикл повторяется снова. Если человек вводит ноль, это будет означать, что программу надо завершить.

Ввод информации от пользователя

В Бейсике ввести в переменную значение с экрана можно с помощью оператора INPUT. Сначала указывается необязательная текстовая подсказка, а потом — имя переменной. Например, оператор

```
INPUT "Введите число: ", x
```

при выполнении напечатает в новой строке подсказку

```
Введите число:
```

и будет ожидать, когда пользователь введет число и нажмет клавишу ENTER. В результате в переменную x запишется новое, введенное с клавиатуры значение.

Главная часть программы

Ввод и редактирование текста программы осуществляется во встроенном редакторе QBasic, правила работы с которым аналогичны правилам работы с большинством известных текстовых редакторов.

Главная часть программы набирается в этом редакторе и должна выглядеть так (комментарии вводить не обязательно):

```
' описание переменной N – числа особей
DIM N AS INTEGER
' начало бесконечного цикла
DO
' ввод числа особей в переменную N
  INPUT "Введите количество особей: ", N
' если введен 0, то
  IF N = 0 THEN
' закончить программу
    END
  END IF
' напечатать результат:
  PRINT "Требуемое число лет: ", Years%(N)
' продолжить цикл с начала
LOOP
```

В тексте используется оператор **END**, который предназначен для немедленно-го завершения работы программы. Операторы, вложенные в цикл и в условные операторы, выделяются отступами, чтобы структура текста была более понятной и наглядной.

Основная, глобальная часть алгоритма реализована. Осталось «спуститься вниз» и запрограммировать функцию **Years%()**, которая в качестве аргумента получает количество особей и возвращает число лет, требуемое для их разведения.

Типы данных в Бейсике

В конце названия функции **Years%** указан символ **%**. Таким образом в Бейсике описывается тип возвращаемого функцией значения. Допустимые символы приведены в таблице.

Потенциальные возможности человека

Объем проекта, строк исходного кода	Тип программы	Время создания	Вероятность успешного завершения	Число программистов
100	Утилиты для временных нужд	1 день	100%	1
1000	Небольшие приложения и дополнения, вносимые в готовые системы	До 1 месяца	100%	1
10 000	Типичная средняя программа, разрабатываемая на заказ	До 6 месяцев	85%	1 (предел возможностей среднего программиста)
100 000	Большинство современных коммерческих автономных и небольших клиент-серверных приложений	1 год	85% для групп, 35% для одиночки	10

Объем проекта, строк исходного кода	Тип программы	Время создания	Вероятность успешного завершения	Число программистов
1 млн	Крупные системы автоматизации	1,5–5 лет	50% для группы, 0% для одиночки	100
10 млн	Операционные системы (Microsoft Windows, IBM VMS), большие военные комплексы. Предел сегояшних возможностей. Стоимость подобной разработки может равняться стоимости большого стадиона или крупного корабля	5–8 лет	35%	До тысячи

Добавление новой функции

В QBasic имеется удобная возможность добавить в программу новую функцию, избежав при этом дополнительного ручного кодирования. Это делает команда **Edit ► New Function** (Правка ► Создать функцию). В появившемся диалоговом окне надо ввести название функции **Years%** и нажать клавишу **ENTER**. Основной текст программы временно пропадет, и появится автоматически сгенерированное описание новой функции:

```
FUNCTION Years%
END FUNCTION
```

Для того чтобы вернуться обратно к главному тексту, а из него — к любой введенной подпрограмме, необходимо использовать клавишу **F2**. При ее нажатии на экран выводится список всех созданных подпрограмм, а в первой строке — имя главного модуля.

Функции **Years%** надо указать список аргументов. В данном случае он будет состоять из одного параметра:

```
FUNCTION Years%(X AS INTEGER)
```

Расчет популяции

Так как для определения нового члена последовательности Фибоначчи требуется знать значения двух предыдущих членов, прежде всего надо описать три локальные переменные **F1**, **F2** и **F3**, хранящие три очередных значения последовательности. Исходно первые три значения 1, 1 и 2 запишутся в переменные **F1**, **F2** и **F3** явно, а в дальнейшем новые значения будут вычисляться программно.

Сам расчет представляет собой условный цикл, который выполняется до тех пор, пока очередное значение не превысит заданное количество особей. Число таких циклов — число лет — будет подсчитываться в локальной переменной-счетчике **YearsNum**, первоначально имеющей значение 3.

```
FUNCTION Years%(X AS INTEGER)
' описание переменных
DIM F1 AS INTEGER, F2 AS INTEGER, F3 AS INTEGER
```

```

DIM YearsNum AS INTEGER
' задание начальных значений
F1 = 1: F2 = 1: F3 = 2: YearsNum = 3
' цикл, пока число кроликов меньше заданного
DO WHILE F3 < X
' определяем новый член последовательности
F1 = F2: F2 = F3
F3 = F1 + F2
' увеличиваем число лет на 1:
YearsNum = YearsNum + 1
' повторяем цикл
LOOP
' в качестве возвращаемого значения
' используется значение переменной YearsNum
Years% = YearsNum
END FUNCTION

```

Сохранение текста программы в файле

После того как текст программы набран, его желательно сохранить в файле, чтобы потом снова обращаться к нему, улучшать, изменять или просто повторно запускать готовую программу.

Сохранение текста программы в файле осуществляется командой **File ► Save** (Файл ► Сохранить), в результате чего на экране показывается диалоговое окно выбора каталога и имени файла. В качестве такого имени можно указать *kroliki*, выбрать нужный каталог и нажать клавишу **ENTER**. По умолчанию к названию *kroliki* припишется расширение *.BAS*. В дальнейшем эту программу можно снова загрузить в QBasic командой **File ► Open** (Файл ► Открыть).

Запуск программы

Для запуска программы надо перейти к ее главной части (с помощью клавиши **F2**) — при этом в самом ее начале автоматически добавится строка с объявлением только что определенной функции:

```
DECLARE FUNCTION Years% (X AS INTEGER)
```

Теперь надо нажать клавишу **F5** (Запуск). Программа начинает работать.

Возможный вариант диалога:

```

Введите количество особей: 10
Требуемое число лет: 7
Введите количество особей: 100
Требуемое число лет: 12
Введите количество особей: 1000
Требуемое число лет: 17
Введите количество особей: 10000
Требуемое число лет: 21
Введите количество особей: 0

```

Первую сотню кроликов надо разводить довольно долго, зато потом их приплод будет увеличиваться стремительными темпами.

20.8. Пример на Паскале. Раскрашивание круга

Постановка задачи

В рабочем окне программы должны находиться: изображение круга, поле ввода с подписью, кнопки **Закрасить** и **Заккрыть**. В поле ввода шестью символами (шестнадцатеричными цифрами от 0 до F) задается новый цвет круга. Первые два символа определяют интенсивность синего цвета, третий и четвертый — интенсивность зеленого, пятый и шестой — интенсивность красного. Чистый синий опишется строкой `ff0000`, чистый зеленый — строкой `00ff00`, чистый красный — строкой `0000ff`, черный — строкой `000000`, белый — строкой `ffffff` и т. д.

В самой программе надо дополнительно проверить, правильно ли введена эта строка. При нажатии на кнопку **Закрасить** цвет круга должен измениться.

Знакомство с Embarcadero Delphi XE

Для работы со средой *Embarcadero RAD Studio XE* необходимо, чтобы она была предварительно установлена на компьютере.

После ее запуска (например, **Пуск** ▶ **Все программы** ▶ **Embarcadero RAD Studio XE** ▶ **Delphi XE**) на экране появится главное окно программы. Программа не создает новый проект с параметрами по умолчанию, поэтому нам нужно создать проект требуемого вида самостоятельно. Для этого сразу после запуска программы следует выполнить команду **File** ▶ **New** (**Файл** ▶ **Новый**) и выбрать в появившемся подменю тип проекта, который вы хотите создать. В данный момент нас устроит проект **VCL Forms Application — Delphi**.

После создания нового проекта экран программы имеет вид, представленный на рис. 20.2.

В верхней части окна программы находится традиционная строка меню и панели инструментов.

В центральной части окна программы расположено окно редактора. В данный момент редактор работает в режиме дизайна (**Design**), в котором производится визуальное проектирование формы будущего приложения. С помощью окна редактора форм можно менять размеры формы, размещать на форме элементы управления, манипулировать ими. Для того чтобы перейти к редактированию исходных текстов программы, следует переключиться в режим **Code** с помощью соответствующего ярлыка в нижней части окна редактора.

В левой нижней части окна программы можно увидеть окно **Object Inspector** (**Инспектор объектов**). Это окно используется для визуальной (без программирования) настройки свойств различных объектов на этапе проектирования.

В правой нижней части окна находится окно **Tool Palette** (**Палитра компонентов**). С ее помощью можно добавлять на проектируемую форму новые элементы управления, выбирая их из групп, представленных на палитре.

Окна **Structure** (**Структура**) и **Project Manager** (**Менеджер проекта**) используются для работы с проектом.

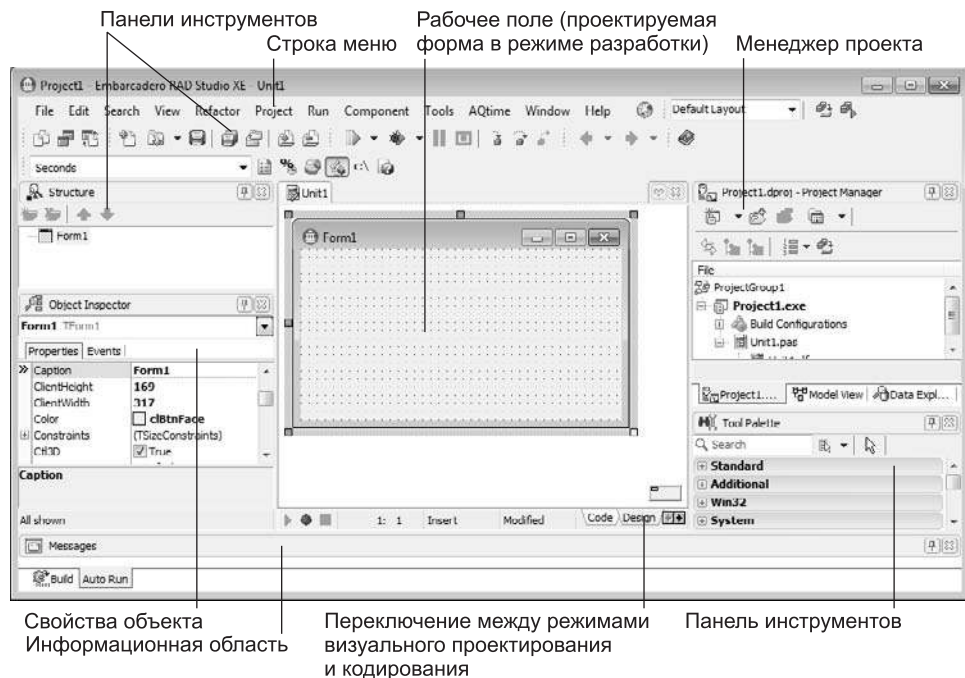


Рис. 20.2. Рабочее окно Embarcadero RAD Studio XE

Заголовок окна

Понять, как работает Инспектор объектов, лучше всего на примере. Пока что автоматически создана только одна пустая форма (называющаяся Form1), но она обладает множеством различных свойств. Заголовок формы (будущего окна программы) задается в свойстве Caption (Заголовок). Чтобы его изменить, надо в Инспекторе объектов найти строку, в левой части которой написано Caption, и в правой части этой строки, небольшом поле ввода, указать новое название, например Раскрашивание. Тут же изменится и заголовок формы в визуальном проектировщике.

Аналогично меняются и любые другие свойства. Сначала в выпадающем списке в верхней части Инспектора выбирается нужный объект (или он выделяется на форме щелчком мыши), затем находится название свойства и в правой части его значение меняется на новое. Изменение может происходить как путем простого ввода нового значения с клавиатуры, так и выбором одного из predetermined значений из списка. В некоторых случаях для редактирования свойства вызывается специальный редактор.

Свойства могут быть многосоставными, например, свойство Font (Шрифт) — слева от названия такого свойства ставится символ «±». Двойным щелчком мыши на его названии оно раскрывается и показывает все свои вложенные подсвойства. Обычно в поле ввода многосоставного свойства можно увидеть кнопку с тремя

точками. Нажатие на эту кнопку открывает отдельное окно для настройки параметров такого свойства.

Размещение компонентов на форме

Прежде всего разместим на форме поле ввода. Для этого на палитре компонентов с помощью закладки выбирается панель **Standard (Стандартная)** и нажимается кнопка с названием **TEdit** (выбран компонент «поле ввода»). Затем надо щелкнуть мышкой на форме, и в месте щелчка появится элемент управления **Edit1**. Его можно перетаскивать по форме и менять размеры.

В свойстве **Text (Содержимое)** этого объекта исходно надо задать пустую строку, чтобы при запуске программы в данном поле ничего не показывалось.

Рядом с полем ввода надо разместить свободное поле с комментарием. Для этого на панели компонентов выбирается компонент, называющийся **TLabel (Подпись)**, и помещается на форме так, как это было сделано с полем ввода. Новый объект автоматически получит название **Label1**. Чтобы указать в нем текст «Цвет:», его надо ввести в свойство **Caption**.

Изменить название любого объекта на форме можно, изменив его свойство **Name** в Инспекторе объектов.

В нижней части формы надо разместить кнопку — компонент **Button (Кнопка)** на панели **Standard (Стандартная)**. Название этой кнопки (**Закрасить**) задается в свойстве **Caption**.

На панели **Additional (Дополнительно)** имеется компонент **TShape (Фигура)**. Этот компонент помещается в центр формы. Он получит название **Shape1** и исходно примет форму квадрата, закрашенного белым цветом. Чтобы превратить его в круг, надо значение свойства **Shape** изменить на **stCircle**.

Теперь осталось только добавить кнопку, закрывающую форму. Это действие стандартное, поэтому в Delphi XE имеется специальный компонент **TBitBtn (Кнопка с картинкой)** на панели **Additional (Дополнительно)**, позволяющий автоматизировать такие действия, не прибегая к программированию.

После размещения такой кнопки на форме (она получит название **BitBtn1**) значение ее свойства **Kind (Вид выполняемого действия)** надо установить в **bkClose (Закрыть окно)**. При этом на кнопке появится изображение стандартной картинки, символизирующей действие закрытия.

В завершение надо изменить заголовок этой кнопки (свойство **Caption**) с английского слова **Close** на русское **Закрыть**.

Теперь добавим на форму еще одну кнопку (компонент **TButton** с панели **Standard**) и введем в ее свойство **Caption** текст «Закрасить». На этом процесс проектирования приложения можно считать законченным.

Сохранение проекта

Перед тем как приступить к программированию, проект надо сохранить. Это действие выполняется командой **File ► Save All (Файл ► Сохранить все)**, после чего сначала выбирается каталог и указывается имя файла, в котором хранится про-

граммное описание (на Паскале) структуры и работы спроектированной формы. Имя файла будет иметь расширение .PAS по умолчанию. Далее система Delphi XE спросит, куда и под каким именем сохранить файл проекта, содержащий всю информацию об используемых формах и модулях (их может быть в одном проекте сколько угодно, но одна форма всегда будет главной) и всевозможные настройки. Название файла проекта **не должно** совпадать с названием файла с исходным текстом программы.

Обработка нажатия кнопки

В создаваемой программе вручную придется запрограммировать фактически только одно событие — нажатие на кнопку Закрасить. Чтобы создать первоначально пустую подпрограмму, вызываемую при нажатии на эту кнопку, надо просто дважды щелкнуть на ней мышкой. При этом система Delphi XE вызовет редактор, автоматически сгенерирует нужный текст и разместит курсор именно в том месте, где можно начать описание нужного алгоритма.

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
end;
```

Обработчик события Нажатие на кнопку Button1 — это обычная подпрограмма, метод класса TForm1 (этот класс описывает главную форму Form1). Единственный параметр Sender характеризует источник сообщения о случившемся событии. Его практически всегда можно игнорировать.

Алгоритм работы данного метода будет следующим. Первоначально надо убедиться, что длина введенной в поле Edit1 строки равна 6 символам и каждый из этих символов — шестнадцатеричная цифра. Если это не так, то выполнение обработчика надо сразу завершить (для этого предназначена стандартная процедура Паскаля Exit, мгновенно завершающая работу текущей подпрограммы).

Если же введенные данные корректны, их надо:

1. Преобразовать в промежуточную строку в формате \$00xxxxxx, где xxxxxx — шесть введенных цифр.
2. Эту строку преобразовать в число, которое будет рассматриваться как цвет.
3. Установить новый цвет круга на основании полученного значения.

Содержимое поля ввода Edit1 хранится в виде строки в его свойстве Text. Доступ к этому свойству осуществляется с помощью конструкции Edit1.Text.

Длина строки определяется стандартной функцией length() со строкой в качестве параметра.

Стандартная функция Pos(), получая две строки как аргументы, проверяет, не содержится ли первая строка во второй, и если содержится, то возвращает номер начальной позиции. В противном случае Pos() возвращает ноль. Эта функция потребуется для определения, все ли символы во введенной строке допустимы.

Стандартная функция UpperCase() преобразует строку к верхнему регистру. Такое преобразование требуется, чтобы разрешить ввод значений цветов на любых регистрах.

Преобразование строки в число выполняет стандартная функция `StrToInt()`.

Объект `Shape1` имеет свойство `Brush` (Кисть для фона), которое, в свою очередь, имеет вложенное свойство `Color` (Цвет заливки). Его и надо в конечном счете изменить. Как только это произойдет, цвет круга в окне автоматически изменится на новый.

```
procedure TForm1.Button1Click(Sender: TObject);
var i: integer;
    s: string;
begin
  // если длина введенной строки не равна 6,
  // то закончить работу
  if length(Edit1.Text) <> 6 then exit;
  // в локальную переменную s заносится строка,
  // содержащая допустимые символы
  s := '0123456789ABCDEF';
  // проверяется каждый символ во введенной строке
  for i := 1 to 6 do
    // если очередной символ не найден в строке s, значит,
    // он недопустим, и работу требуется прекратить
    if pos(UpperCase(Edit1.Text[i]), s) = 0 then exit;
  // все нормально – в переменной s
  // готовим промежуточную строку
  s := '$00'+Edit1.Text;
  // Устанавливаем значение цвета заливки круга равным
  // числу, преобразованному из строки в переменной s
  Shape1.Brush.Color := StrToInt(s);
end;
```

Запуск программы

Программа запускается нажатием на клавишу `F9`. Так как Delphi XE — это компилирующая система, сначала автоматически выполнится компиляция, и только потом программа запустится. Задавая различные строки (`FF0FFF`, `abcdef`, `987654` и т. п.), можно наглядно увидеть соответствующие им цвета.

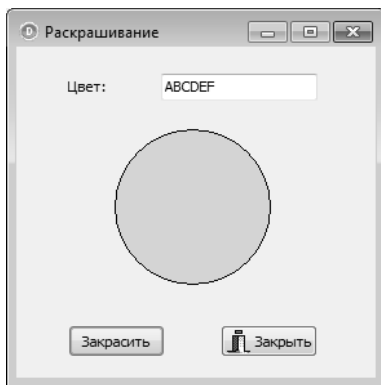


Рис 20.3. Программа закраски в работе

20.9. Пример на C++. Рисование графиков

Система *Microsoft Visual Studio 2010* во многом схожа с рассмотренной средой Delphi. Здесь применяются те же принципы визуального проектирования программных продуктов. Для разработки проектов можно применять языки программирования C++, C#, Visual Basic. При желании можно использовать и другие языки.

Постановка задачи

Существует таблица, содержащая зависимость значения функции от аргумента. Надо нарисовать график, отражающий эту зависимость.

Решение задачи

Для решения задачи запустим *Microsoft Visual Studio 2010* (Пуск ► Все программы ► *Microsoft Visual Studio 2010* ► *Microsoft Visual C++ 2010*) и создадим новый проект командой File ► New ► Project (Файл ► Новый ► Проект). В появившемся окне выберем *Windows Forms Application*, зададим имя проекта (например, Graph) в поле Name (Имя) и нажмем OK.

Внешний вид окна разработки проекта очень похож на тот, который мы рассматривали, изучая среду Delphi. Здесь есть окна редактирования кода и визуальной разработки формы, окно для работы со структурой проекта (*Solution Explorer*), окно для выбора элементов управления, размещаемых на форме (*Toolbox*), окно для настройки свойств выбранного элемента управления (*Properties*).

Для изменения стандартного заголовка формы на «Построение графика» следует отредактировать свойство Text.

Разместим на форме элемент управления Chart (Диаграмма) из группы Data (Данные). Этот комплексный элемент управления может применяться для построения графиков и диаграмм различных типов. Элемент получит имя (Name) chart1.

Разместим на форме элемент управления Button (кнопка) из раздела Common Controls. Кнопка получит имя button1. Зададим ее свойство Text как Построить график.

Выполним двойной щелчок по кнопке, откроется редактор кода, содержащий процедуру обработчика события нажатия на кнопку.

Разместим в обработчике следующий код:

```
double data[10];  
data[0]=1.3;  
data[1]=3.4;  
data[2]=2.3;  
data[3]=5.7;  
data[4]=0.4;  
data[5]=7.1;  
data[6]=3.2;  
data[7]=4.3;  
data[8]=1.9;
```

```
data[9]=5.8;
chart1->Series["Series1"]->Name="Результаты измерений";
chart1->Series["Результаты измерений"]->ChartType=System::Windows::Forms::DataVisualization::Charting::SeriesChartType::Line;
for(int i=0;i<10;i++)
{
    chart1->Series["Результаты измерений"]->Points->AddXY(i+1,data[i]);
}
button1->Enabled=false;
```

Здесь командой

```
double data[10];
```

мы создаем массив для хранения данных, которые должны быть отображены на графике.

Командами:

```
data[0]=1.3;
```

и другими мы заполняем массив данными, которые должны быть изображены на графике.

Командой

```
chart1->Series["Series1"]->Name="Результаты измерений";
```

мы меняем имя (Name) ряда данных Series1 на Результаты измерений. Это имя автоматически отобразится на графике в качестве подписи ряда данных, его же мы будем использовать при обращении к серии программно.

Команда

```
chart1->Series["Результаты измерений"]->ChartType=System::Windows::Forms::DataVisualization::Charting::SeriesChartType::Line;
```

позволяет программно поменять тип графика, установив его в Line, то есть в линейный график. По умолчанию элемент управления отображает данные в виде гистограммы. Надо отметить, что этот и другие параметры можно настроить как интерактивно, до запуска программы, так и программно, во время работы.

Блок команд

```
for(int i=0;i<10;i++)
{
    chart1->Series["Результаты измерений"]->Points->AddXY(i+1,data[i]);
}
```

предназначен для организации цикла от 0 до 9. В каждой итерации цикла мы добавляем к коллекции точек данных временного ряда Результаты измерений новую точку, в качестве координаты X которой используем значение $i + 1$ (нулевое i соответствует первому результату измерения и т. д.), в качестве Y — соответствующее значение массива data.

Последней командой:

```
button1->Enabled=false;
```

мы блокируем кнопку, то есть делаем ее недоступной для дальнейших нажатий пользователем.

После завершения редактирования кода проект надо сохранить, выполнить компиляцию и запустить, нажав клавишу F5. На рис. 20.4 можно видеть результаты рисования графика.

В дальнейшем описанную программу можно развивать по различным направлениям. Например, предусмотреть ввод данных для построения графика пользователем с клавиатуры во время выполнения программы, либо устанавливать значения Y путем вызова внешней функции, вычисляющей их в зависимости от значения X , получив, таким образом, универсальный инструмент для построения простых графиков.

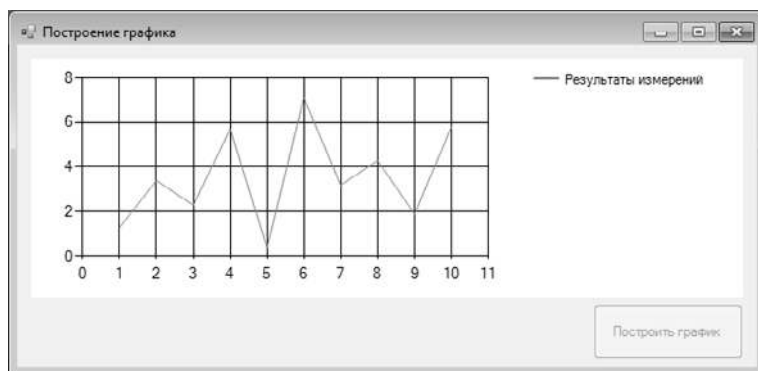


Рис. 20.4. Результат построения графика

Практические задания по программированию

Задание 1

Дано натуральное число. Составить программу, которая представляет данное число в виде суммы квадратов натуральных чисел, содержащей минимальное число слагаемых. Например:

$$\begin{aligned}9 &= 3^2; \\12 &= 2^2 + 2^2 + 2^2; \\23 &= 3^2 + 3^2 + 2^2 + 1^2.\end{aligned}$$

Задание 2

Дан массив, содержащий N элементов.

Написать подпрограммы, выполняющие следующие действия:

- ☐ перестановку элементов массива в обратном порядке;
- ☐ вычисление суммы $A[1] + A[2]*A[2] + A[3]*A[3]*A[3] \dots$;
- ☐ определение элементов массива, разность модулей которых имеет наибольшее значение;
- ☐ определение значения, которое встречается среди элементов массива максимальное число раз, и вычисление количества таких вхождений;
- ☐ упорядочение элементов массива по возрастанию.

Задание 3

Дан двумерный массив, содержащий $N \times N$ элементов.

Написать подпрограммы, выполняющие следующие действия:

- ☐ вычисление среднего арифметического для элементов каждой строки массива;
- ☐ замену нулями всех элементов, расположенных на главной диагонали матрицы;
- ☐ определение наибольшего элемента и его положения в массиве.

Задание 4

Дана текстовая строка.

Написать подпрограммы, выполняющие следующие действия:

- ☐ подсчет количества слов в строке (в качестве границ слов рассматриваются пробелы);
- ☐ подсчет количества цифр в строке;
- ☐ определение десятичного числа, которому соответствует строка, если она представляет запись этого числа в шестнадцатеричной системе;
- ☐ проверку соответствия содержимого строки правилам записи идентификаторов языков программирования.

Сергей Витальевич Симонович
Информатика. Базовый курс: Учебник для вузов
3-е издание
Стандарт третьего поколения

Заведующий редакцией
Руководитель проекта
Ведущий редактор
Художественный редактор
Корректор
Верстка

А. Кривцов
А. Юрченко
Ю. Сергиенко
Л. Адуевская
В. Листова
Е. Егорова

ООО «Мир книг», 198206, Санкт-Петербург, Петергофское шоссе, 73, лит. А29.
Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3005 — литература учебная.
Подписано в печать 18.02.11. Формат 70х100/16. Усл. п. л. 51,600. Тираж 5000. Заказ 0000.
Отпечатано по технологии StP в ОАО «Печатный двор» им. А. М. Горького.
197110, Санкт-Петербург, Чкаловский пр., 15.