

62-50
К60
СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

Е. А. Колмыкова
И. А. Кумскова

ИНФОРМАТИКА

10-е издание

ОБЩЕОБРАЗОВАТЕЛЬНЫЕ
ДИСЦИПЛИНЫ



УЧЕБНОЕ ПОСОБИЕ



62-25
СРЕДНЕЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

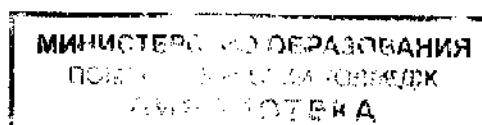
Е. А. КОЛМЫКОВА, И. А. КУМСКОВА

ИНФОРМАТИКА

*Допущено
Министерством образования Российской Федерации
в качестве учебного пособия для студентов образовательных
учреждений среднего профессионального образования*

10-е издание, стереотипное

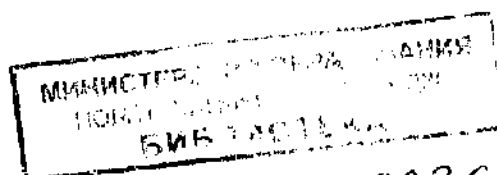
96336



Москва
Издательский центр «Академия»
2012

УДК 621.391(075.32)
ББК 32.81я723
К17

Рецензенты:
преподаватель информатики Московского колледжа
градоостроительства и предпринимательства *О. С. Антонова*;
преподаватель кафедры «Информатизация банковской деятельности»
Московского банковского института *Е. В. Михеева*



К17 Колмыкова Е. А.
Информатика : учеб. пособие для студ. учреждений сред.
проф. образования / Е. А. Колмыкова, И. А. Кумскова. — 10-е
изд., стер. — М. : Издательский центр «Академия», 2012. —
416 с.

ISBN 978-5-7695-9469-4

Характерной особенностью учебного пособия является систематиза-
ция рассматриваемых вопросов. Особое внимание уделено изучению ос-
нов логики, моделированию и формализации задач, использованию ин-
формационных технологий и объектному программированию на языке
Visual Basic.

Для студентов учреждений среднего профессионального образования.

УДК 621.391(075.32)
ББК 32.81я723

*Оригинал-макет данного издания является собственностью
Издательского центра «Академия», и его воспроизведение любым способом
без согласия правообладателя запрещается*

© Колмыкова Е. А., Кумскова И. А., 2005
© Образовательно-издательский центр «Академия», 2005
© Оформление. Издательский центр «Академия», 2005

ISBN 978-5-7695-9469-4

ПРЕДИСЛОВИЕ

Во второй половине XX в. человечество вступило в новый этап своего развития — начался переход от индустриального общества к информационному. Процесс, обеспечивающий этот переход, получил название информатизации. Это процесс создания, развития и всеобщего применения информационных средств и технологий, обеспечивающих достижение и поддержание уровня информированности всех членов общества, необходимого и достаточного для кардинального улучшения качества труда и условий жизни в обществе. При этом информация становится важнейшим стратегическим ресурсом общества и занимает ключевое место в экономике, образовании и культуре.

Неизбежность информатизации общества обусловлена резким возрастанием роли и значения информации. Информационное общество характеризуется высокоразвитой информационной сферой, которая включает в себя деятельность человека по созданию, переработке, хранению, передаче и накоплению информации.

Современная научная картина мира совмещает в себе два основных аспекта: вещественно-энергетический и информационный. Информатика как наука должна обеспечивать формирование информационной картины мира, систематическое изучение информационных процессов в природе, обществе и технике.

В настоящее время существует несколько трактовок понятия «информатика». Первое направление рассматривает информатику как науку, связанную с информационными технологиями и компьютеризованными системами. Второе направление, в основном под влиянием трудов В. И. Сифорова, сводит ее к учению об «информации вообще» — информологии. Это направление по существу развивает теорию информации К. Шеннона. Третье направление ставит в центр информатики семантические (содержательные) стороны информации.

Разнообразие понимания предмета информатики как науки, отсутствие оригинальных понятий (например, *информация*), не пересекающихся с понятиями других наук кибернетического цикла, свидетельствуют о том, что информатика переживает этап накопления и осмысления теоретического материала. Во многих учебниках доминируют прикладные разработки, решения частных вопросов, практически односторонние суждения.

В процессе изучения курса «Информатика» учащиеся должны овладеть современными методами научного познания (моделирование и формализация, компьютерный эксперимент и т. д.), научиться создавать и исследовать модели из различных предметных областей для формирования системного и алгоритмического (операционного) мышления.

Цели данного учебного пособия:

- осуществить четкое изложение основных понятий и современных подходов к информатике как самостоятельной науке естественно-научного направления;
- на основании анализа объекта и предметной области информатики изложить фундаментальные категории и аксиомы, которые служат стержнем информатики как науки;
- дать математические основы информатики как базисный инструмент для решения прикладных задач;
- изложить методологию функционирования программ и приложений на основе объектно-ориентированного подхода;
- дать необходимые знания, привить умения и навыки в области информационных технологий (поиск и систематизация информации, использование информационных ресурсов глобальной компьютерной сети Интернет);
- установить семантические связи между материалом различных глав и тем учебного пособия.

Авторы многих учебников и учебных пособий по информатике выделяют в отдельные разделы теоретическую информатику, информационные технологии и программирование. Такое разделение материала, на наш взгляд, является достаточно условным, так как в каждом разделе можно выделить как теоретические, так и прикладные аспекты.

В предлагаемом учебном пособии материал размещен в восьми главах и приложениях.

Материал гл. 1 содержит основные понятия и определения информатики. Анализируются свойства и виды информации. Приводятся сведения и исторические факты развития информатики и информационных технологий. Излагаются общие вопросы, связанные с осмыслением роли и значения информации в развитии общества.

В гл. 2 рассматриваются основные принципы построения современных персональных компьютеров (ПК). Изложение материала начинается с изучения систем счисления, используемых для представления числовой информации, форматы данных, кодирование и декодирование символьной, графической и звуковой информации в ПК.

Излагаются теоретические основы алгебры логики, анализируется связь между алгеброй логики и двоичным кодированием. Даются общие принципы организации и работы компьютера.

В гл. 3 приводятся сведения о технических средствах информационных процессов и устройствах, входящих в состав современного компьютера и подключаемых к нему. Понимая, что технические характеристики устаревают очень быстро, основной упор сделан на принципы работы устройств. Рассматриваются вопросы организации компьютерных сетей, сетевых протоколов, доступа в глобальную сеть Интернет и обеспечения безопасности данных. Программное обеспечение (ПО) — неотъемлемая часть компьютерной системы, поэтому в гл. 3 подобран материал по классификации ПО, назначению и основным функциям операционных систем.

Более прикладной характер носят гл. 4 и 5. В гл. 4 рассматриваются вопросы практической работы в операционной системе Windows (на примере Windows XP) в соответствии с принципами объектного подхода; раскрываются возможности наиболее распространенных сервисных программ и описываются практические приемы работы с ними, особое внимание уделяется программам-архиваторам и антивирусным программам.

В гл. 5 дается общая характеристика процессов сбора, передачи, обработки и хранения информации. Рассматриваются также актуальные вопросы защиты информации от несанкционированного доступа к ней. В качестве инструментария в настоящее время используются следующие виды программных продуктов: текстовые процессоры, настольные издательские системы, электронные таблицы, системы управления базами данных, графические редакторы, мультимедийные системы, электронная почта и т. д. Цель гл. 5 — дать основное представление о возможностях обработки информации различного вида и познакомить учащихся с методами использования инструментария информационных технологий.

В гл. 6 рассматриваются вопросы создания моделей различных видов. Приводятся характеристики информационных моделей, и раскрывается суть операции формализации. Вводится понятие информационной технологии решения задачи, и анализируются этапы решения задачи на компьютере. Этот материал является одним из ключевых в курсе «Информатика», но он достаточно сложен для восприятия, поэтому сопровождается большим количеством примеров.

Целью изучения материала гл. 7 является выработка основ алгоритмического мышления. Рассматриваются основные приемы алгоритмизации, приводятся примеры разработки алгоритмов решения типовых задач, даются подробные объяснения методики их решения, приводятся схемы алгоритмов. Изложенный материал активно используется в гл. 8.

В гл. 8 изложены основы программирования. Дается классификация языков и технологий программирования, излагается концепция объектно-ориентированного подхода. Кроме классических

основ программирования излагается введение в среду проектирования Visual Basic 6, которая обладает богатым арсеналом инструментов, позволяющих создавать разнообразные приложения.

Материал, представленный в учебном пособии, не является абсолютно полным. Более того, приведенные технические характеристики вскоре устареют, появятся новые программные продукты и технологии. Однако, по мнению авторов, основное достоинство данного учебного пособия заключается в последовательном и четком изложении материала: информация — технические средства — информационный ресурс — информационная технология — модель — программа-приложение.

Авторы выражают искреннюю признательность С. В. Воробьевой и А. А. Цветковой за большую помощь в процессе подготовки книги к изданию, а также всем коллегам, которые помогали в работе над данным учебным пособием.

Глава 1

ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

В этой главе изложены общие вопросы, связанные с осмыслением роли и значения информации в развитии общества. Вы узнаете причины возникновения, цели и задачи информатики, поймете, почему она призвана сыграть решающую роль в формировании информационного общества, и сможете ответить на следующие вопросы: Что такое информатика? Для чего она нужна? Какие задачи она решает?

Понятие «информация» занимает центральное место в информатике, поэтому изучение теоретических основ этой науки начинается с методов и моделей количественной оценки информации.

Работа с информацией сегодня требует, чтобы человек свободно владел компьютерными технологиями, знал разные методы обработки информации на компьютере, умел правильно формализовывать задачи, свободно работал в среде информационных систем. Поэтому чем выше уровень развития страны, тем более интенсивно проходит процесс, получивший название «информатизация». Его конечной целью является формирование общества, в котором живут и работают информационно грамотные люди, умело использующие компьютерные системы и технологии.

1.1. ВВЕДЕНИЕ В ИНФОРМАТИКУ

Термин «информатика» (от *фр.* information — информация + *automatique* — автоматика) означает «информационная автоматика». Широко распространен также англоязычный вариант этого термина *Computer science* — компьютерная наука.

Информатика — это наука, изучающая структуру и общие свойства информации, а также закономерности и методы ее создания, хранения, поиска, передачи и преобразования с использованием компьютерных технологий.

Выделение информатики как самостоятельной сферы человеческой деятельности связано, в первую очередь, с развитием компьютерной техники (причем основная заслуга в данном случае принадлежит микропроцессорной технике, появление которой в середине 1970-х гг. послужило началом четвертой информационной революции).

В 1978 г. Международный научный конгресс официально закрепил за понятием «информатика» области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты компьютеризации — массового внедрения компьютерной техники во все области жизни людей.

Информатика — комплексная научная дисциплина с широчайшим диапазоном применения. Ее приоритетными направлениями являются:

- теория информации, изучающая процессы, связанные с передачей, приемом, преобразованием и хранением информации;
- разработка вычислительных систем и программного обеспечения;
- математическое моделирование, методы вычислительной и прикладной математики и их применение к фундаментальным и прикладным исследованиям в различных областях знаний;
- методы искусственного интеллекта, моделирующие методы логического и аналитического мышления в интеллектуальной деятельности человека (логический вывод, обучение, понимание речи, визуальное восприятие и др.);
- системный анализ, изучающий методологические средства, используемые для подготовки и обоснования решений по сложным проблемам различного характера;
- биоинформатика, изучающая информационные процессы в биологических системах;
- социальная информатика, изучающая процессы информатизации общества;
- методы машинной графики, анимации, средства мультимедиа;
- телекоммуникационные системы и сети, в том числе глобальные компьютерные сети, объединяющие все человечество в единое информационное сообщество;
- разнообразные приложения, охватывающие производство, науку, образование, медицину, торговлю, сельское хозяйство и другие виды хозяйственной и общественной деятельности.

В нашей стране трактовка термина «информатика» утвердилась с момента принятия соответствующего решения в 1983 г. на сессии ежегодного собрания Академии наук СССР об организации нового отделения информатики, вычислительной техники и автоматизации. Информатика трактовалась как «... комплексная научная и инженерная дисциплина, изучающая все аспекты разработки, проектирования, создания, оценки, функционирования основанных на ЭВМ систем переработки информации, их при-

менения и воздействия на различные области социальной практики».

Методы информатики применимы всюду, где существует возможность описания объекта, явления, процесса с помощью информационных моделей, поэтому информатика нацелена на разработку общих методологических принципов построения информационных моделей (см. гл. 6).

Российский академик А. А. Дородницын выделяет в информатике три неразрывно и существенно связанные части — технические, программные и алгоритмические средства.

Технические средства, или аппаратура компьютеров, в английском языке обозначаются словом «Hardware», которое буквально переводится как «твердые изделия» или «железо».

Программные средства — совокупность всех программ, используемых компьютерами, и область деятельности по их созданию и применению. Для обозначения программных средств используется слово «Software» (буквально — мягкие изделия или программное обеспечение), которое подчеркивает равнозначность самой машины и программного обеспечения, а также способность программного обеспечения модифицироваться, приспосабливаться и развиваться.

Для обозначения алгоритмических средств применяют термин «Brainware» (от *англ.* brain — интеллект).

Программированию задачи всегда предшествует разработка способа ее решения в виде последовательности действий, ведущих от исходных данных к искомому результату, иными словами, разработка алгоритма решения задачи.

Часто путают понятия «информатика» и «кибернетика». В чем же их сходство и различие?

Основная концепция, заложенная Н. Винером в кибернетику, связана с разработкой теории управления сложными динамическими системами в разных областях человеческой деятельности. Кибернетика существует независимо от наличия или отсутствия компьютеров.

С одной стороны, информатика занимается изучением процессов преобразования и создания новой информации более широко, практически не решая задачи управления различными объектами, как кибернетика. Поэтому может сложиться впечатление об информатике как о более емкой дисциплине, чем кибернетика. С другой стороны, информатика не занимается решением проблем, не связанных с использованием компьютерной техники, что, несомненно, сужает ее, казалось бы, обобщающий характер. Между этими двумя понятиями провести четкую границу не представляется возможным.

Роль информатики в развитии общества чрезвычайно велика. С ней связано начало революции в области накопления, передачи

и обработки информации. Эта революция, следующая за революциями в овладении веществом и энергией, затрагивает и коренным образом преобразует не только сферу материального производства, но и интеллектуальную, духовную сферы жизни общества, формирует информационную культуру общества.

Прогрессивное увеличение возможностей компьютерной техники, развитие информационных сетей, создание новых информационных технологий приводят к значительным изменениям во всех сферах жизни общества (в производстве, науке, образовании, медицине и т.д.).

Контрольные вопросы

1. Что означает термин «информатика»?
2. Какие сферы человеческой деятельности затрагивает информатика?
3. Назовите составные части информатики.
4. Является ли информатика наукой или прикладной дисциплиной?

1.2. ИНФОРМАЦИЯ В РЕАЛЬНОМ МИРЕ

Принципы всеобщего энергоинформационного взаимодействия в природе были известны человеку разных исторических эпох. В первую очередь, это были наблюдения и накопление информации об окружающем мире. Собранные знания и опыт человечество использовало для защиты и выживания в сложных условиях, для создания орудий труда. Большие объемы разнообразных накопленных сведений способствовали развитию научно-технического прогресса, с помощью которого люди научились управлять веществом и различными видами энергии. Но и сегодня, как и раньше, кроме вещества и энергии огромную роль в жизни человечества играет еще одна составляющая — информация, неразрывно связанная с ними.

Информация — это совокупность разнообразных данных, сведений, сообщений, знаний, умений и опыта, необходимых кому-либо.

Длительное время ею владели и пользовались лишь отдельные представители цивилизации, устанавливая собственную власть и могущество. Сегодня информация имеет общественное значение — это огромные публичные библиотеки, хранящие знания о литературе, искусстве, медицине, образовании, достижениях научной мысли в технике; видео- и аудиотеки; всемирная информационная компьютерная сеть. Информация так же, как вещество и энергия, стала предметом производства и распространения, обрела свойства и функции товара.

В широком смысле информация — это общенаучное понятие, включающее в себя обмен сведениями между людьми, обмен

сигналами между живой и неживой природой, людьми и устройствами.

Наряду с понятием «информация» в информатике часто употребляется понятие «данные».

Данные — это результаты наблюдений над объектами и явлениями, которые по каким-то причинам не используются, а только хранятся. Как только данные начинают использовать в каких-либо практических целях, они превращаются в информацию.

Исходя из этого можно определить информацию как «используемые данные».

Предположим, что у вас дома постоянно работает телевизор. Обратите внимание на то, как вы воспринимаете передаваемые по нему сведения. Далеко не все служит для вас информацией. Это зависит от вашего интереса к тому или иному сообщению. Только в том случае когда некое сообщение принесло вам что-то новое, можно говорить о получении информации. В остальных случаях сообщение сводится к простому набору данных. Информация несет человеку новые знания об объектах, процессах, явлениях.

Процесс обработки информации очень сложен и зависит от множества факторов как объективного, так и субъективного характера. На протяжении всей своей жизни человек постоянно участвует во всевозможных информационных процессах.

Информационный процесс — процесс, в результате которого осуществляется прием, передача (обмен), преобразование и использование информации.

С середины XX в. интенсивность информационных процессов существенно увеличилась. Лавинообразный поток информации, хлынувший на человека, уже не воспринимается в полном объеме, ориентироваться в нем становится все труднее. Подчас оказывается проще заново создать какой-либо продукт, нежели разыскать аналог, сделанный ранее. Поэтому требуется постоянное обновление и совершенствование способов, помогающих человеку воспринимать, преобразовывать, хранить и использовать информацию.

Всегда ли компьютер выглядел так, как в настоящее время? Чем обусловлены возможности вычислительной техники? Рассмотрим и сопоставим достижения в области вычислительной техники, в результате которых происходила смена поколений компьютеров (электронно-вычислительных машин (ЭВМ)).

Первое поколение компьютеров — с середины 1940-х до конца 1950-х гг. Элементная база — электронные лампы. ЭВМ отличаются большими габаритными размерами, потреблением большого количества энергии, малой скоростью действия, низкой надежностью. Программирование ведется в машинных кодах.

Второе поколение компьютеров — с конца 1950-х до середины 1960-х гг. Элементная база — полупроводниковые элементы. По

сравнению с ЭВМ предыдущего поколения улучшены все технические характеристики. Для программирования используются алгоритмические языки.

Третье поколение компьютеров — с середины 1960-х до конца 1970-х гг. Элементная база — интегральные схемы, многослойный печатный монтаж. Значительное уменьшение габаритных размеров ЭВМ, повышение их надежности, увеличение производительности. Доступ с удаленных терминалов.

Четвертое поколение компьютеров — с конца 1970-х гг. по настоящее время. Элементная база — микропроцессоры, большие интегральные схемы. Улучшены технические характеристики. Массовый выпуск персональных компьютеров. Направления развития: мощные многопроцессорные вычислительные системы с высокой производительностью; создание дешевых микроЭВМ; опытные разработки интеллектуальных компьютеров; внедрение во все сферы компьютерных сетей и их объединение, распределенная обработка данных, повсеместное использование компьютерных информационных технологий.

Таким образом, появление нового типа ЭВМ было обусловлено изобретением новой элементной базы. С позиций информатики четвертую информационную революцию можно связать с появлением ЭВМ четвертого поколения — персонального компьютера, с успехом позволяющего решать проблему хранения и передачи информации на качественно новом уровне. Информационная революция, произошедшая в 1970-х гг., привела к тому, что человеческая цивилизация в конце XX в. оказалась в состоянии перехода от индустриальной фазы своего развития к информационной.

Результатом процесса информатизации является создание информационного общества, в котором главную роль играют интеллект и знания. Для каждой страны ее движение от индустриального этапа развития к информационному определяется степенью информатизации общества.

Увеличивается число людей, профессионально занимающихся сбором, накоплением, обработкой, распространением и хранением информации.

Перспективы перехода к информационному обществу влекут за собой проблемы социального, правового и технического характера, так как внедрение новых компьютерных и электронных технологий создает необходимость получения людьми соответствующего образования.

Кроме того, информация все в большей мере становится стратегическим ресурсом государства, производительной силой и дорогим товаром. Это не может не вызывать стремления государств, организаций и отдельных граждан получить преимущества за счет овладения информацией, недоступной оппонентам, а также за

счет нанесения ущерба информационным ресурсам противника (конкурента) и защиты своих информационных ресурсов, т.е. информационной безопасности.

Значимость обеспечения безопасности государства в информационной сфере сформулирована в принятой в сентябре 2000 г. «Доктрине информационной безопасности Российской Федерации».

В Российской Федерации принят ряд указов, постановлений, таких как «Об авторском праве и смежных правах», «О правовой охране программ для ЭВМ и баз данных», «О правовой охране топологий интегральных схем».

Рассмотрим основные положения Закона РФ «Об информации, информатизации и защите информации» от 20.02.95 № 24-03, который является базовым юридическим документом, открывающим путь к принятию дополнительных нормативных законодательных актов для успешного развития информационного общества. С его помощью удалось частично решить вопросы правового регулирования на информационном рынке: защиты прав и свобод личности; угроз и ущерба, связанных с искажением, порчей, уничтожением персональной информации.

В законе определены цели и основные направления государственной политики в сфере информатизации. Информатизация определяется как важное новое стратегическое направление деятельности государства. Указано, что государство должно заниматься формированием и реализацией единой государственной научно-технической и промышленной политики в сфере информатизации.

Закон создает условия для включения России в международный информационный обмен, предотвращает бесхозяйственное отношение к информационным ресурсам, обеспечивает информационную безопасность и права юридических и физических лиц. В нем определяются комплексное решение проблемы организации информационных ресурсов и правовые положения по их использованию.

Закон закладывает юридические основы гарантий прав граждан на информацию. Он направлен на урегулирование важнейшего вопроса экономической реформы — формы, права и механизма реализации собственности на накопленные информационные ресурсы и технологические достижения. Обеспечена защита собственности в сфере информационных систем и технологий, что способствует формированию цивилизованного рынка информационных услуг, систем, технологий, средств их обеспечения.

Не менее остро стоит вопрос информационного противоборства на уровне организаций и отдельных граждан. Об этом свидетельствуют многочисленные попытки криминальных элементов получить контроль над компьютерными технологиями для извле-

чения материальной выгоды. Можно привести в качестве примера случаи шантажа английских фирм преступной международной группой. За 1993—1996 гг. преступники получили 400 млн фунтов стерлингов. Жертвам приходилось выплачивать до 13 млн фунтов стерлингов одновременно после демонстрации шантажистами своих возможностей по отмене всех сделок или получению доступа к новейшим разработкам фирм. Деньги переводились в банки, расположенные в офшорных зонах, откуда преступники снимали их в считанные минуты.

Проблема обеспечения информационной безопасности на всех уровнях может быть решена успешно только в том случае, если создана и функционирует комплексная система защиты информации, охватывающая весь жизненный цикл компьютерных систем, от разработки до утилизации, и всю техническую цепочку сбора, хранения, обработки и выдачи информации.

Контрольные вопросы

1. Чем отличаются понятия «информация» и «данные»?
2. Что такое информационный процесс? Приведите примеры.
3. Дайте краткую характеристику поколений ЭВМ.
4. Что послужило отправной точкой для перехода от индустриального общества к информационному?
5. Что входит в понятие «информационная безопасность»?
6. Что входит в понятие «информационная культура»?
7. Является ли наше общество информационным? Как вы представляете информационное общество?
8. Чем отличаются понятия «компьютеризация» и «информатизация»?
9. Всегда ли потеря информации является результатом злого умысла?
10. Какой закон обеспечивает информационную безопасность и права юридических и физических лиц?

1.3. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Одним из основных направлений, по которому осуществляется информатизация общества, является повсеместное использование информационных систем (ИС) и информационных технологий (ИТ). Без этого невозможен переход от индустриального общества к информационному.

Введем базовое с точки зрения информатики понятие «система».

Система — любой объект, который одновременно рассматривается и как единое целое, и как совокупность разнородных объектов, объединенных для достижения определенного результата.

Системы различаются между собой как по составу, так и по цели функционирования. Например, системой является образова-

тельное учреждение, коммерческая фирма, автомобиль, компьютер и еще множество различных объектов, которые в зависимости от цели можно рассматривать и как единое самостоятельное целое, и как совокупность нескольких объектов, взаимодействующих между собой.

Информационные системы обеспечивают сбор, хранение, обработку, поиск и выдачу информации.

В широком смысле информационной системой можно назвать любую организационную структуру, задача которой состоит в работе с информацией. Примеры таких структур: библиотека, справочная служба железных дорог, пресса (редакция газеты, телецентр, радиостудия). В этом смысле информационными системами являются все подразделения управленческой структуры предприятия: бухгалтерия, отдел кадров, отдел научно-технической информации и др. Все эти службы существовали и до появления компьютеров, существуют и сейчас. Разница заключается в том, что раньше они использовали «бумажные» технологии работы с информацией, простые средства механизации обработки данных, а сейчас все шире используют компьютеры. В дальнейшем речь будет идти только о компьютерных информационных системах.

Информационная система — взаимосвязанная совокупность средств, методов и персонала, участвующих в обработке информации.

В работе ИС на равных участвуют как технические и программные средства, так и человек. Только в результате их взаимодействия возможна обработка первичной информации и получение информации нового качества.

Любая ИС может действовать по правилам разомкнутой или замкнутой системы управления.

В *разомкнутой* ИС (рис. 1.1, а) получаемая потребителем информация используется произвольно. От потребителя в информационную систему ничего не поступает. В этом случае говорят о работе системы в автономном разомкнутом режиме, когда цель ее функционирования не определяется потребителем. Примером работы разомкнутой ИС служит компьютеризированная справочная библиотечная система каталогов, которая обеспечивает любого читателя информацией по интересующей его тематике, или справочная система аэропорта (вокзала), информирующая пассажиров о вылете самолетов.

В *замкнутой* ИС (рис. 1.1, б), напротив, существует тесная связь между ее структурой и потребителем. В этом случае ИС ориентирована на конкретного потребителя и на его цели, что достигается за счет введения в ее структуру канала обратной связи, которому передается реакция потребителя на полученную им информацию. Например, идея замкнутой ИС воплощена при организации работы железнодорожной кассы. Сначала кассир обеспечивается необ-

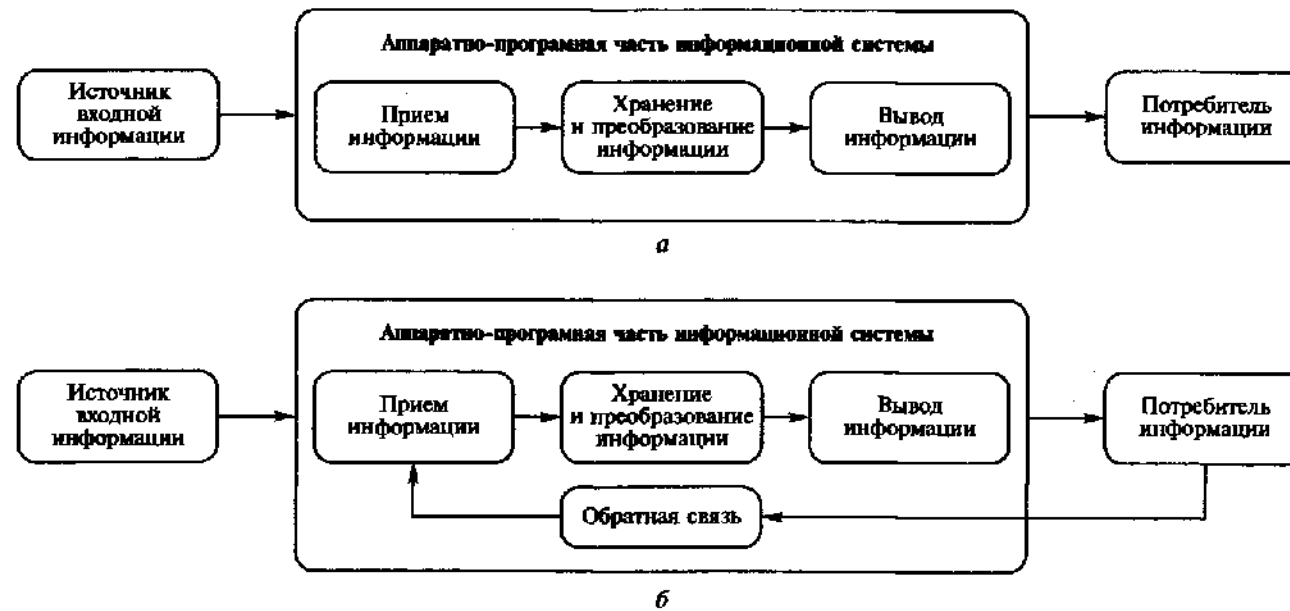


Рис. 1.1. Виды информационных систем:
 а — разомкнутая; б — замкнутая система

ходимой информацией о наличии билетов. Как только очередной билет продан, кассир вводит сообщение об этом в компьютер. Соответствующая программа производит отметку о проданном билете. В замкнутой системе обратная связь от потребителя реализована в виде сведений о проданных билетах.

Многочисленные и разнообразные ИС, которые существуют сегодня, можно классифицировать по разным признакам.

По используемой технической базе различают ИС:

- простейшие, работающие на одном компьютере. Вся информация сосредоточена в памяти этой машины, и на ней же функционирует все программное обеспечение системы;
- на базе локальной сети. В такой системе информация может передаваться по сети между разными пользователями; разные части общедоступных данных могут храниться на разных компьютерах сети;
- на базе глобальных компьютерных сетей. Все известные службы Интернета можно рассматривать как таковые. Наиболее масштабной из них является World Wide Web. Имеется также множество так называемых корпоративных ИС.

По назначению, т.е. по выполняемым функциям, различают ИС:

- информационно-справочные, или информационно-поисковые, — наиболее старый и традиционный вид ИС. Основная цель использования таких систем — оперативное получение ответов на запросы пользователей в диалоговом режиме. Характерным свойством информационно-поисковых систем (ИПС) является большой объем хранимых данных, их постоянная обновляемость. Обычно пользователь желает получить ответ на свой запрос быстро, поэтому качество системы во многом определяется скоростью поиска данных и выдачи ответа. При работе ИПС не используются сложные методы обработки данных. Хранилище информации, с которой работает ИПС, называется базой данных. Примером справочной системы является ИПС крупной библиотеки, позволяющая определить наличие в библиотеке нужной книги или произвести подборку литературы по заданной тематике. Поисковые серверы Интернета — это информационно-справочные системы сетевых ресурсов;

- управляющие системы, основное назначение которых — выработка управляющих решений. Управляющие системы бывают полностью автоматическими или автоматизированными:

системы автоматического управления (САУ), работающие без участия человека. Это системы управления техническими устройствами, производственными установками, технологическими процессами;

автоматизированные системы управления (АСУ), которые можно назвать человеко-машинными системами. В них компьютер выступает в роли помощника человека-управля-

БИБЛИОТЕКА

ющего. В АСУ задача компьютера состоит в оперативном предоставлении человеку необходимой информации для принятия решения. При этом компьютер может выполнять достаточно сложную обработку данных на основании заложенных в него математических моделей. Это могут быть технологические или экономические расчеты;

- обучающие системы на базе ЭВМ, простейшим вариантом которых является обучающая программа на ПК, с которой пользователь работает в индивидуальном режиме. Существует множество таких программ практически по всем курсам профессионального обучения. В локальной сети можно организовывать обучение с элементами взаимодействия учащихся между собой, используя соревновательную форму или форму деловой игры;

- системы дистанционного обучения, работающие в глобальных сетях, — наиболее сложные и масштабные обучающие системы. Дистанционное образование называют образованием XXI в. Уже существуют дистанционные отделения во многих ведущих вузах страны, формируется международная система дистанционного образования. Такие системы открывают доступ к качественному образованию для всех людей, независимо от их места жительства, возраста, возможных физических ограничений. Высокоскоростные системы связи в сочетании с технологией мультимедиа позволяют организовывать обучение в режиме реального времени (online), проводить дистанционные лекции, семинары, конференции, принимать зачеты и экзамены;

- экспертные системы — системы, основанные на моделях знаний из определенных предметных областей. Экспертные системы относятся к разделу информатики, который называется «Искусственный интеллект». Экспертная система заключает в себе знания высококвалифицированного специалиста в определенной предметной области и используется для консультаций, помощи в принятии сложных решений, решения плохо формализуемых задач. Примерами задач, которые решаются с помощью экспертных систем, являются установление диагноза болезни, определение причин неисправности сложной техники (например, космического корабля), выдача рекомендаций по ликвидации неисправности, определение вероятных последствий принятого управляющего решения и т.д.

Рассмотрим, как могут быть реализованы функции информационной системы.

Информация является одним из самых ценных ресурсов общества наряду с такими природными богатствами, как нефть, газ, полезные ископаемые и др. Следовательно, процесс переработки информации по аналогии с процессом переработки материальных ресурсов можно определить как технологию (от *гр. techne* — искусство, мастерство, умение).

Технология материального производства определяется совокупностью средств и методов обработки, изготовления, изменения состояния, свойств, формы сырья или материала. Технология изменяет качество или первоначальное состояние материала (рис. 1.2).

Информационная технология — совокупность средств и методов обработки и передачи первичной информации для получения информации нового качества о состоянии объекта, процесса или явления.

Информационные ресурсы — это идеи человечества и указания по реализации этих идей, накопленные в форме, позволяющей их воспроизводство (книги, статьи, патенты, диссертации, научно-исследовательская и опытно-конструкторская документации, технические переводы, данные о передовом производственном опыте и др.).

Информационные ресурсы (в отличие от всех других видов ресурсов — трудовых, энергетических, минеральных и т.д.) растут тем быстрее, чем больше их расходуют.

Цель информационной технологии — производство информации для ее последующего анализа и принятия на его основе решения по выполнению какого-либо действия.

До второй половины XIX в. существовала «ручная» информационная технология, инструментарий которой составляли перо, чернильница, бухгалтерская книга.

С конца XIX в. применялась механическая технология, инструментарий которой составляли пишущая машинка, телефон, фонограф, почта.

В 40—60-е гг. XX в. использовалась электрическая технология, инструментарий которой составляли большие ЭВМ и соответствующее программное обеспечение, электрические пишущие машинки, копировальные аппараты, портативные магнитофоны.

С начала 1970-х гг. внедряется электронная технология, основным инструментарием которой становятся большие ЭВМ и создаваемые на их базе автоматизированные системы управления,

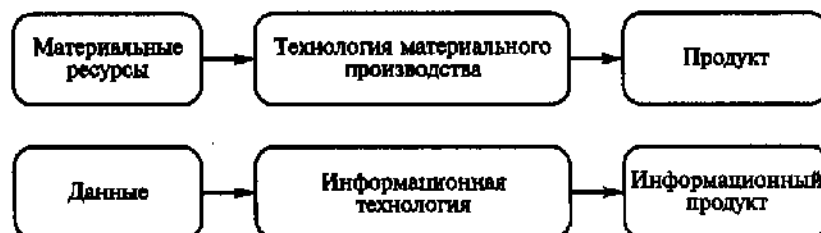


Рис. 1.2. Информационная технология как аналог технологии переработки материальных ресурсов

оснащенные широким спектром базовых и специализированных программных комплексов.

С середины 1980-х гг. появилась компьютерная технология, основным инструментарием которой является персональный компьютер с большим количеством стандартных программных продуктов разного назначения.

Рассмотрим соотношение между информационными технологиями и системой.

Информационная технология представляет собой процесс, состоящий из четко регламентированных правил выполнения различных операций с данными, хранящимися в компьютере.

Информационная система — это среда, равноправными элементами которой являются: персонал, компьютеры, компьютерные сети, программные продукты, базы данных, различного рода технические и программные средства связи и т. д.

Реализация функций ИС не возможна без знания ориентированной на нее информационной технологии. Информационная технология может существовать и вне сферы ИС.

Контрольные вопросы

1. Дайте определение информационной системы.
2. Перечислите типы информационных систем.
3. Что такое разомкнутая ИС? Приведите примеры.
4. Что такое замкнутая ИС?
5. Сформулируйте цели информационной технологии.
6. Возможна ли ИС без применения информационной технологии?
7. Какие процессы происходят в ИС?
8. Приведите как можно больше примеров различных видов информационных технологий.

1.4. ИНФОРМАЦИЯ И ЕЕ СВОЙСТВА

Возможность эффективного использования информации обуславливается ее качественными характеристиками или свойствами:

- **понятность.** Информация должна быть понятной всем участникам обмена информацией. Например, человек — существо социальное, для общения с другими людьми должен обмениваться с ними информацией, причем обмен информацией всегда происходит на определенном языке (русском, английском и т. д.), поэтому участники дискуссии должны владеть тем языком, на котором ведется общение;

- **достоверность.** Информация должна быть достоверной, т. е. она должна отражать истинное положение дел. Недостоверная информация может привести к неправильному пониманию или при-

нятию неправильных решений. Достоверная информация со временем может стать недостоверной, так как она обладает свойством устаревать, т.е. перестает отражать истинное положение дел;

- полнота. Информация должна быть полной, если ее достаточно для понимания и принятия решений. Как неполная, так и избыточная информация сдерживает принятие решений и может повлечь ошибки;

- ценность. Ценность информации зависит от того, насколько она важна для решения задачи, а также от того, насколько в дальнейшем она найдет применение в каких-либо видах деятельности человека;

- своевременность. Информация должна быть своевременной — только в этом случае она может принести ожидаемую пользу. Одинаково нежелательны как преждевременная подача информации (когда она еще не может быть усвоена), так и подача информации с задержкой.

1.4.1. Измерение информации

Для того чтобы рассмотреть участие информации в информационном процессе, необходимо ввести количественные характеристики информации, т.е. научиться ее измерять.

В теоретической информатике информация рассматривается как знания, т.е. процесс систематического научного познания окружающего мира приводит к накоплению информации в виде знаний (научных теорий, фактов и т.д.).

Процесс познания можно наглядно изобразить в виде расширяющегося круга знания (такой способ придумали еще древние греки). Вне этого круга лежит область незнания, а окружность является границей между знанием и незнанием. Парадокс состоит в том, что чем большим объемом знаний обладает человек и чем шире круг его знаний, тем больше он ощущает недостаток знаний и тем больше граница его незнания, мерой которого в этой модели является длина окружности.

Например, объем знаний выпускника школы гораздо больше, чем объем знаний первоклассника или пятиклассника, однако и граница его незнания также существенно больше. Действительно, первоклассник совершенно ничего не знает о законах физики, химии или экономики и его это не смущает, тогда как выпускник школы, например, при подготовке к экзамену по физике может обнаружить, что есть законы, которых он не знает или не понимает. Можно считать, что ученик, получая информацию, уменьшает неопределенность знания (расширяет круг знания). Подход к информации как к мере уменьшения неопределенности знания позволяет количественно измерять информацию, что чрезвычай-

но важно для информатики, учитывая, что она имеет дело с процессами передачи и хранения информации.

Информацию можно рассматривать как набор сообщений.

Сообщение — это форма представления каких-либо сведений в виде речи, текста, изображения, цифровых данных, графиков, таблиц и т.п.

Сообщение, в свою очередь, состоит в возможности наступления некоторых событий, каждое из которых может иметь свою вероятностную характеристику, т.е. степень возможности наступления этого события.

Пример 1. В закрытом ящике лежат два шара — черный и белый. Вытаскиваем один шар. Перед вытаскиванием существовала неопределенность нашего знания, так как возможны два события: «черный шар» или «белый шар». После того как шар вытащен, наступает полная определенность: событие «черный шар», тогда в ящике остался белый, и наоборот.

Вытаскивание одного из двух шаров приводит к уменьшению неопределенности нашего знания в 2 раза.

Рассмотрим понятие «вероятность».

Если N — это общее число возможных исходов какого-то процесса (например, вытаскивание шара), а из них интересующее нас событие (например, вытаскивание белого шара) может произойти k раз, то вероятность этого события p можно определить по формуле

$$p = k/N.$$

Вероятность выражается в долях единицы. Для примера 1 вероятность вытаскивания как белого, так и черного шара равна $1/2$, т.е. они равновероятны.

Вероятность достоверного события равна 1 (из 50 белых шаров вытащен белый шар); вероятность невозможного события равна 0 (из 50 белых шаров вытащен черный шар).

Пример 2. В ящике лежат четыре шара — три черных и один белый. Вытаскиваем один шар. Его цвет, скорее всего, будет черным, но может быть и белым.

Посчитаем вероятность вытаскивания белого и черного шара:

$$p_{\text{бел}} = 1/4 = 0,25; p_{\text{черн}} = 3/4 = 0,75.$$

Информация содержится в сообщении о цвете вытащенного шара. Какая информация в примере 2 ценнее: «белый шар» или «черный шар»? Конечно, информация о том, что вытащили белый шар, т.е. этим сообщением получено полное знание, что в коробке остались только черные шары.

Информация о том, что вытащили черный шар, тоже уменьшает неопределенность знания (после этого события в ящике осталось три шара — один белый и два черных), но не дает полного знания, какой шар может быть вытащен следующим.

Качественную связь между вероятностью события и количеством информации в сообщении об этом событии можно выразить так: чем меньше вероятность некоторого события, тем больше информации содержит сообщение об этом событии.

Количественная зависимость между вероятностью события p и количеством информации в сообщении о нем i выражается формулой

$$i = \log_2(1/p), \quad (1)$$

или

$$2^i = 1/p. \quad (2)$$

Если событие достоверно, его вероятность равна 1, то оно неинформативно, т.е. количество информации в нем равно 0, но чем меньше вероятность какого-то события, тем большую ценность имеет информация об этом событии и тем больше будет значение i .

Единицей измерения количества информации является бит (от англ. bit — binary digit — двоичная цифра).

1 бит — количество информации, необходимой для различения двух равновероятных событий.

Вернемся к примеру 1.

Пусть x — количество информации в сообщении о том, что вытащен белый шар. Тогда

$$2^x = 1/0,5 \Rightarrow 2^x = 2 \Rightarrow x = 1 \text{ бит},$$

т.е. мы доказали, что сообщение об одном событии из двух равновероятных несет 1 бит информации.

Количество информации можно рассчитать методами Р.Хартли и К.Шеннона.

Формула Р.Хартли:

$$i = \log_2 N.$$

Американский инженер Р.Хартли в 1928 г. рассматривал процесс получения информации как выбор одного сообщения из конечного наперед заданного множества из N равновероятных сообщений, а количество информации i , содержащейся в выбранном сообщении, определял как двоичный логарифм N .

Допустим, нужно угадать одно число из набора чисел от 1 до 100. По формуле Р.Хартли можно вычислить, какое количество информации для этого требуется:

$$i = \log_2 100 = 6,644.$$

Таким образом, сообщение о верно угаданном числе содержит количество информации, приблизительно равное 6,644 бит.

Приведем другие примеры равновероятных сообщений:

- при бросании монеты: «выпала решка», «выпал орел»;
- на странице книги: «число букв четное», «число букв нечетное».

Определим теперь, являются ли равновероятными сообщения «первой выйдет из дверей здания женщина» и «первым выйдет из дверей здания мужчина». Однозначно ответить на этот вопрос нельзя. Все зависит от того, о каком именно здании идет речь. Если это, например, станция метро, то вероятность выйти из дверей первым одинакова для мужчины и женщины, а если это военная казарма, то для мужчины эта вероятность значительно выше, чем для женщины. Для задач такого рода американский ученый К. Шеннон предложил в 1948 г. другую формулу определения количества информации, учитывающую возможную неодинаковую вероятность сообщений в наборе.

Формула Шеннона:

$$i = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_N \log_2 p_N),$$

где p_i — вероятность того, что именно i -е сообщение выделено в наборе из N сообщений.

Заметим, что если вероятности $p_1, \dots, p_N$ равны, то каждая из них равна $1/N$ и формула К. Шеннона превращается в формулу Р. Хартли.

Помимо двух рассмотренных подходов к определению количества информации существуют и другие. Важно помнить, что любые теоретические результаты применимы лишь к определенному кругу случаев, очерченному первоначальными допущениями.

Пример 3. В мешке лежат 64 монеты. Сообщение о том, что достали золотую монету, несет 4 бит информации. Сколько золотых монет было в мешке?

Дано: $N = 64$; $i_{\text{зол}} = 4$.

Найти: $k_{\text{зол}}$.

Сообщение о том, что достали золотую монету, несет 4 бит информации, следовательно:

$$2^4 = 1/p_{\text{зол}}.$$

Отсюда можно найти вероятность вытаскивания золотой монеты:

$$p_{\text{зол}} = 1/16.$$

С другой стороны, $p_{\text{зол}} = k_{\text{зол}}/N$, следовательно, $k_{\text{зол}} = Np_{\text{зол}} = 64/16 = 4$.

Ответ: число золотых монет — 4.

Пример 4. В ящике лежат 8 черных шаров и 24 белых. Сколько информации несет сообщение о том, что достали черный шар?

Дано: $k_{\text{черн}} = 8$; $k_{\text{бел}} = 24$.

Найти: $i_{\text{черн}}$.

$$N = k_{\text{бел}} + k_{\text{черн}} = 32;$$

$$p_{\text{черн}} = k_{\text{черн}}/N = 8:32 = 1/4;$$

$$2^{i_{\text{черн}}} = 1/p_{\text{черн}} = 4;$$

$$i_{\text{черн}} = 2 \text{ бит.}$$

Ответ: сообщение о том, что достали черный шар, несет 2 бит информации.

Пример 5. При игре в кости используется кубик с шестью гранями. Сколько бит информации получает игрок при каждом бросании кубика?

Выпадение каждой грани кубика равновероятно и равно $1/6$. Поэтому количество информации от одного результата бросания находится из уравнения

$$2^i = 6.$$

Решая это уравнение по формуле (1): $i = \log_2 6$, получаем ответ: $i = 2,585$ бит. Решение примера 5 можно получить, воспользовавшись табл. 1.1, в которой N — общее число равновероятных событий; i — количество информации, бит.

Теперь рассмотрим понятие информации с точки зрения ее передачи и хранения.

Информация передается в виде сообщений от некоторого источника информации к ее приемнику посредством канала связи между ними. Источник посылает передаваемое сообщение, которое кодируется в передаваемый сигнал. Этот сигнал посылается по каналу связи. В результате в приемнике появляется принимаемый сигнал, который декодируется и становится принимаемым сообщением (рис. 1.3).

Информация может существовать в виде следующих сигналов:

- тексты, рисунки, чертежи, фотографии;
- световые или звуковые сигналы;
- радиоволны;
- электрические и нервные импульсы;
- магнитные записи;
- жесты и мимика;
- запахи и вкусовые ощущения;
- хромосомы, посредством которых передаются по наследству признаки и свойства организмов, и т.д.

Например, сообщение, содержащее информацию о прогнозе погоды, передается приемнику (телезрителю) от источника (специалиста-метеоролога) посредством канала связи — телевизионной передающей аппаратуры и телевизора.

Таблица 1.1

Количество информации в сообщении об одном
из N равновероятных событий

N	I	N	I	N	I	N	I
1	0,00000	17	4,08764	33	5,04439	49	5,61471
2	1,00000	18	4,16993	34	5,08746	50	5,64386
3	1,58496	19	4,24793	35	5,12928	51	5,67243
4	2,00000	20	4,32193	36	5,16993	52	5,70044
5	2,32193	21	4,39232	37	5,20945	53	5,72792
6	2,58496	22	4,45943	38	5,24793	54	5,75489
7	2,80735	23	4,52356	39	5,28540	55	5,78136
8	3,00000	24	4,58496	40	5,32193	56	5,80735
9	3,16993	25	4,64386	41	5,35755	57	5,83289
10	3,32193	26	4,70044	42	5,39232	58	5,85798
11	3,45943	27	4,75489	43	5,42626	59	5,88246
12	3,58496	28	4,80735	44	5,45943	60	5,90689
13	3,70044	29	4,85798	45	5,49185	61	5,93074
14	3,80735	30	4,90689	46	5,52356	62	5,95420
15	3,90689	31	4,95420	47	5,55459	63	5,97728
16	4,00000	32	5,00000	48	5,58496	64	6,00000

Вы научились определять количество информации, которое содержится в сообщениях, уменьшающих неопределенность знаний. Такой подход рассматривает информацию с точки зрения содержания, ее понятности и новизны для человека (например, в опыте по вытаскиванию шара одинаковое количество информации содержится и в коротком сообщении «черный», и в длинной фразе «мы вытащили черный шар»). Однако при хранении или передаче информации с помощью технических устройств первое сообщение займет меньше места, чем второе. Поэтому целесообразно рассматривать информацию как последовательность знаков некоего алфавита (букв, цифр, кодов цвета точек и т.д.), тогда объем информации в сообщении — это число символов в этом сообщении.

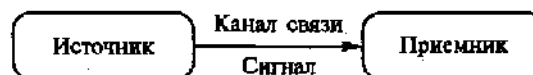


Рис. 1.3. Передача данных

Системы счисления можно рассматривать как формальные языки, имеющие алфавит (цифры) и позволяющие не только именовать и записывать объекты (числа), но и выполнять над ними арифметические операции по строго определенным правилам (см. гл. 2).

Некоторые алфавиты используют в качестве знаков не буквы и цифры, а другие символы (например, музыкальные ноты, изображения элементов электрических или логических схем, дорожные знаки, точки и тире и др.).

Если считать, что появление символов в сообщении равновероятно, то по формуле $2^i = 1/p$ можно рассчитать количество информации, которое несет каждый символ.

Информационная емкость знаков зависит от их числа в алфавите: чем больше их число, тем большее количество информации несет один знак.

Так, информационная емкость буквы в русском алфавите, если не использовать букву «ё», составляет: общее число букв — 32, следовательно, $p_{\text{буквы}} = 1/32$, и тогда по формуле (2) количество информации i составляет: $2^i = 32$, т. е. $i = 5$ бит.

Вероятностный метод применим и для алфавитного подхода к измерению информации, заключенной в тексте. Известно, что разные символы (буквы алфавита, знаки препинания и др.) встречаются в тексте с разной частотой и, следовательно, имеют разную вероятность.

Значит, измерять информационный вес каждого символа в тексте так, как это делалось раньше (в предположении равновероятности), нельзя.

Пример 6. В зашифрованном сообщении встречаются 4 буквы (X, Y, Z, W), точка и пробел; его объем 1000 знаков, из них: букв X — 400, букв Y — 100, букв Z — 200, букв W — 150, точек — 50, пробелов — 100. Какой объем информации содержит зашифрованное сообщение?

Поскольку объем сообщения достаточно большой, то можно допустить, что вычисленная по ней частота встречаемости в тексте каждого символа алфавита характерна для любого текста. Подсчитаем частоту встречаемости каждого символа во всем тексте шифровки (т. е. вероятность) и информационные веса символов:

$$\begin{aligned} \text{буква } X &\Rightarrow 400 : 1000 = 0,4 \Rightarrow i_X = \log_2(1 : 0,4) = 1,321928; \\ \text{буква } Y &\Rightarrow 100 : 1000 = 0,1 \Rightarrow i_Y = \log_2(1 : 0,1) = 3,1928; \\ \text{буква } Z &\Rightarrow 200 : 1000 = 0,2 \Rightarrow i_Z = \log_2(1 : 0,2) = 2,321928; \\ \text{буква } W &\Rightarrow 150 : 1000 = 0,15 \Rightarrow i_W = \log_2(1 : 0,15) = 2,736966; \\ \text{точка} &\Rightarrow 50 : 1000 = 0,05 \Rightarrow i_{\text{точка}} = \log_2(1 : 0,05) = 4,321928; \\ \text{пробел} &\Rightarrow 100 : 1000 = 0,1 \Rightarrow i_{\text{пробел}} = \log_2(1 : 0,1) = 3,321928. \end{aligned}$$

Общий объем информации в зашифрованном сообщении вычислим как сумму произведений информационного веса каждого символа на число повторений этого символа в книге:

$$i = i_x \cdot n_x + i_y \cdot n_y + i_z \cdot n_z + i_w \cdot n_w + i_{\text{точка}} \cdot n_{\text{точка}} + i_{\text{пробел}} \cdot n_{\text{пробел}} = \\ = 1,321928 \cdot 400 + 3,1928 \cdot 100 + 2,321928 \cdot 200 + 2,736966 \cdot 150 + \\ + 4,321928 \cdot 50 + 3,321928 \cdot 100 = 2284,184 \text{ бит.}$$

Ответ: общий объем информации в сообщении составляет 2284,184 бит.

1.4.2. КОДИРОВАНИЕ ИНФОРМАЦИИ

В процессе обмена информацией часто приходится производить операции кодирования и декодирования информации. При вводе знака алфавита в компьютер путем нажатия соответствующей клавиши на клавиатуре происходит его кодирование, т.е. преобразование в компьютерный код. При выводе знака на экран монитора или принтер происходит обратный процесс — декодирование, когда из компьютерного кода знак преобразуется в графическое изображение.

Кодирование/декодирование — это операция преобразования знаков или групп знаков одной знаковой системы в знаки или группы знаков другой знаковой системы.

Способ кодирования зависит от цели, ради которой оно осуществляется (например, желания сократить объем, скрыть информацию, представить информацию в ином виде).

В компьютере для внутримашинного представления данных и команд используется единица измерения количества информации — бит.

Битом называют наименьшую «порцию» памяти, необходимую для хранения одного из двух знаков: 0 и 1.

Бит — слишком мелкая единица измерения количества информации. На практике чаще применяется более крупная единица измерения — байт, равная 8 бит. Именно 8 бит требуется для того, чтобы закодировать любой из 256 символов алфавита клавиатуры компьютера ($256 = 2^8$).

Широко используются также более крупные производные единицы информации:

- 1 Килобайт (Кбайт) = 1024 байт = 2^{10} байт;
- 1 Мегабайт (Мбайт) = 1024 Кбайт = 2^{20} байт;
- 1 Гигабайт (Гбайт) = 1024 Мбайт = 2^{30} байт.

В последнее время в связи с увеличением объемов обрабатываемой информации входят в употребление еще более крупные производные единицы информации:

- 1 Терабайт (Тбайт) = 1024 Гбайт = 2^{40} байт;
- 1 Петабайт (Пбайт) = 1024 Тбайт = 2^{50} байт.

Контрольные вопросы

1. Какие существуют показатели качества информации?
2. Что можно делать с информацией?
3. Как можно оценить объем информации в сообщении?
4. Что определяет 1 бит в вычислительной технике и в теории информации?
5. Какое количество информации будет получено при игре в рулетку с 64 секторами?
6. Вероятность первого события составляет 0,5, второго и третьего — 0,25. Какое количество информации получится после реализации одного из них?
7. В классе 30 человек. За контрольную работу по математике получено 6 пятерок, 15 четверок, 8 троек и 1 двойка. Какое количество информации в сообщении о том, что Иванов получил четверку?
8. «Преподаватель держит в руке лист бумаги с напечатанным текстом». Является ли это сообщение информацией?
9. Книги на русском и китайском языках содержат одинаковое число знаков. В какой книге содержится большее количество информации?
10. Для чего используется кодирование информации?

Глава 2



ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ И РАБОТЫ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

Персональный компьютер стал обязательным атрибутом на любом рабочем месте. Это основная техническая база любой информационной технологии.

В этой главе рассмотрены принципы и организация представления информации в устройствах вычислительной техники (в том числе, в ПК). Основы построения и функционирования были заложены задолго до появления современного ПК, на этапе разработки и конструирования первых электронно-вычислительных машин. В этой главе рассмотрены основные системы счисления, используемые в вычислительной технике, числовая система ПК, представление чисел, кодирование и декодирование символьной, графической, звуковой информации, форматы данных.

2.1. ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

В вычислительных устройствах информация (числовая, текстовая, графическая, звуковая и т.д.) кодируется в последовательность электрических импульсов: есть импульс (1), нет импульса (0), т.е. в последовательность нулей и единиц.

Цифры 1 и 0 можно рассматривать как два равновероятных состояния (события). При записи двоичной цифры реализуется выбор одного из двух возможных состояний (одной из двух цифр) и, следовательно, она несет количество информации, равное 1 бит (см. подразд. 1.4).

Количество информации в битах равно числу цифр двоичного машинного кода.

2.1.1. Системы счисления

Система счисления — математическая модель или знаковая система, в которой числа записываются по определенным правилам с помощью цифр.

Существуют непозиционные и позиционные системы счисления.

В *непозиционных* системах счисления вес цифры (т.е. тот вклад, который она вносит в значение числа) не зависит от ее позиции в записи числа. Так, в римской системе счисления в числе XXXII (тридцать два) вес цифры X в любой позиции равен десяти.

В *позиционных* системах счисления (табл. 2.1) вес каждой цифры изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число. Например, в десятичном числе 757,7 первая семерка означает 7 сотен, вторая — 7 единиц, а третья — 7 десятых долей единицы. Сама же запись числа 757,7 означает сокращенную запись суммы $700 + 50 + 7 + 0,7 = 7 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0 + 7 \cdot 10^{-1} = 757,7$.

Любая позиционная система счисления характеризуется основанием q .

Основание позиционной системы счисления q — это число цифр, используемых для изображения чисел в данной системе счисления.

Приняв за основание число 10, получим хорошо знакомую десятичную систему счисления. Число 60 является основанием древней вавилонской шестидесятеричной системы счисления, к которой восходит деление часа на 60 мин и угла на 360° . Традицию считать дюжинами (в году 12 мес, в сутках два периода по 12 ч, в футе 12 дюймов) распространили англосаксы. В Китае широко использовалась пятеричная система счисления.

Какое же преимущество имеют позиционные системы счисления перед непозиционными и почему именно они получили наибольшее распространение? Ответ на этот вопрос достаточно прост: в позиционной системе счисления намного легче производить арифметические операции.

Например, необходимо сложить числа 29,7 и 8,2.

В римской системе счисления (XXIX,VII + VIII,II) такую операцию провести достаточно трудно, если не сказать невозможно.

Таблица 2.1

Позиционные системы счисления

Основание q	Система счисления	Цифры (алфавит)
2	Двоичная	0, 1
3	Троичная	0, 1, 2
...	...	...
8	Восьмеричная	0, 1, 2, 3, 4, 5, 6, 7
10	Десятичная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16	Шестнадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

В десятичной системе счисления:

$$\begin{array}{r} 29,7 \\ + 8,2 \\ \hline 37,9 \end{array}$$

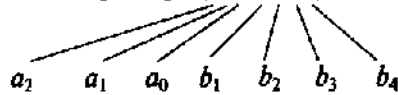
Существует алгоритм выполнения арифметических операций (операции «в столбик»), при котором при переполнении разряда числа, равного основанию системы счисления q (для десятичной системы счисления $q = 10$), единица переносится в старший разряд (в нашем примере это 1, полученная после сложения 9 и 8), а при вычитании и делении при необходимости единица занимает-ся в старшем разряде и переходит в младший количеством еди-ниц, равным q .

2.1.2. Перевод чисел из системы счисления с основанием q в десятичную

Любое вещественное число в системе счисления с основанием q можно представить набором коэффициентов (цифр) в следу-ющем виде:

$$(a_n a_{n-1} \dots a_1 a_0, b_1 b_2 \dots b_m)_q.$$

Например: $(127,0354)_8$



Для перевода числа в десятичную систему счисления исполь-зуется формула

$$a_n \cdot q^n + a_{n-1} \cdot q^{n-1} + \dots + a_1 \cdot q^1 + a_0 \cdot q^0 + b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + \dots + b_m \cdot q^{-m}.$$

Примеры.

$$4718,63_{10} = 4 \cdot 10^3 + 7 \cdot 10^2 + 1 \cdot 10^1 + 8 \cdot 10^0 + 6 \cdot 10^{-1} + 3 \cdot 10^{-2} = 4000 + 700 + 10 + 8 + 0,6 + 0,03 = 4718,63;$$

$$1001,1_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 8 + 1 + 0,5 = 9,5;$$

$$7764,1_8 = 7 \cdot 8^3 + 7 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0 + 1 \cdot 8^{-1} = 3584 + 448 + 48 + 4 + 0,125 = 4084,125;$$

$$3AF,4_{16} = 3 \cdot 16^2 + A \cdot 16^1 + F \cdot 16^0 + 4 \cdot 16^{-1} = 768 + 160 + 15 + 0,25 = 943,25.$$

В последнем примере вместо букв шестнадцатеричной системы счисления используются десятичные числа:

$$A - 10; B - 11; C - 12; D - 13; E - 14; F - 15.$$

2.1.3. Перевод чисел из десятичной системы счисления в систему счисления с основанием q

Перевод вещественного числа производится по двум алгоритмам: алгоритм 1 — для целой части числа; алгоритм 2 — для его дробной части.

Алгоритм 1. Последовательно выполнять целочисленное деление целой части числа и получаемых целых частных на основание системы счисления q до тех пор, пока не получится частное, меньшее делителя (q).

Полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления.

Составить число в новой системе счисления, записывая его, начиная с последнего остатка.

Пример. Перевести десятичное число 173_{10} в восьмеричную систему счисления.

Число	Делитель	Целочисленный остаток
173	8	5
21	8	5
2	8	2

Получаем: $173_{10} = 255_8$.

Пример. Перевести десятичное число 173_{10} в шестнадцатеричную систему счисления.

Число	Делитель	Целочисленный остаток
173	16	13
10	16	10

Получаем: $173_{10} = AD_{16}$.

Пример. Перевести десятичное число 11_{10} в двоичную систему счисления.

Число	Делитель	Целочисленный остаток
11	2	1
5	2	1
2	2	0
1	2	1

Получаем: $11_{10} = 1011_2$.

Алгоритм 2. Последовательно умножать правильную дробь и получаемые дробные части произведений на основание системы q до тех пор, пока дробная часть произведения не станет равной нулю или не будет достигнута требуемая точность представления числа.

Полученные целые части произведений, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления.

Составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

Пример. Перевести число $0,65625_{10}$ в восьмеричную систему счисления.

0,	65625
	$\times 8$
5,	25000
	$\times 8$
2,	00000

Получаем: $0,65625_{10} = 0,52_8$.

Пример. Перевести число $0,65625_{10}$ в шестнадцатеричную систему счисления.

0,	65625
	$\times 16$
10,	25000
(A)	$\times 16$
8,	00000

Получаем: $0,65625_{10} = 0,48_{16}$.

Перевод вещественных чисел, т.е. чисел, содержащих целую и дробную части, осуществляется в два этапа. Отдельно переводится целая часть, отдельно — дробная. В итоговой записи полученного числа целая часть отделяется от дробной запятой.

Пример. Перевести число $17,25_{10}$ в двоичную систему счисления.

Переводим целую часть:

Число	Делитель	Остаток
17	2	1
8	2	0
4	2	0
2	2	0
1	2	1

Получаем: $17,25_{10} = 10001,01_2$.

Переводим дробную часть:

0,	25
	$\times 2$
0,	50
	$\times 2$
1,	00

2.1.4. Системы счисления, используемые в компьютере

В компьютере используют двоичную систему счисления для представления информации, потому что она имеет ряд преимуществ перед другими системами счисления:

- для ее реализации нужны технические устройства с двумя устойчивыми состояниями (а не с десятью, как в десятичной системе счисления). Например: электромагнитное реле (замкнуто/разомкнуто), которое широко использовалось в конструкциях первых ЭВМ; участок поверхности магнитного носителя информации (намагничен/размагничен); участок поверхности лазерного диска (отражает/не отражает); триггер, который может устойчиво находиться в одном из двух состояний;
- широко используется в оперативной памяти компьютера;
- представление информации посредством только двух состояний надежно и помехоустойчиво;
- возможно применение аппарата булевой алгебры для выполнения логических преобразований информации (см. гл. 3);
- двоичная арифметика намного проще десятичной.

Недостаток двоичной системы счисления — быстрый рост числа разрядов, необходимых для записи чисел.

Человеку очень трудно воспринимать многоразрядные числа, т.е. числа, записанные в двоичном коде, а для компьютера разрядность числа не имеет большого значения, так как современные компьютеры обрабатывают за один такт работы процессора более 64 двоичных разрядов.

Перевод чисел из десятичной системы счисления в двоичную и наоборот выполняет машина, однако программисты часто используют восьмеричную и шестнадцатеричную системы счисления на этапах отладки программ и просмотра содержимого файлов в режиме машинных кодов.

Числа в этих системах счисления читаются почти так же легко, как десятичные, требуют соответственно в три (восьмеричная) и в четыре (шестнадцатеричная) раза меньше разрядов, чем в двоичной системе счисления (числа 8 и 16 — соответственно третья и четвертая степени числа 2).

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему счисления очень прост; достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) для восьмеричной системы счисления или тетрадой (четверкой цифр) для шестнадцатеричной системы счисления.

Примеры.

$$537,1_8 = \underbrace{101}_5 \underbrace{011}_3 \underbrace{111}_7 \underbrace{001}_1; \quad 1A3, F_{16} = \underbrace{1}_{1} \underbrace{1010}_A \underbrace{0011}_3 \underbrace{1111}_F.$$

Чтобы перевести число из двоичной системы счисления в восьмеричную или шестнадцатеричную, его нужно разбить влево и вправо от запятой на триады или тетрады и каждую такую группу заменить соответствующей восьмеричной или шестнадцатеричной цифрой.

Примеры.

$$10101001, 10111_2 = \underbrace{10}_2 \underbrace{101}_5 \underbrace{001}_1, \underbrace{101}_5 \underbrace{111}_6_2 = 251,56_8;$$

$$10101001, 10111_2 = \underbrace{1010}_A \underbrace{1001}_9, \underbrace{1011}_B, \underbrace{1000}_8_2 = A9, B8_{16}.$$

2.1.5. Арифметические операции в позиционных системах счисления

Рассмотрим основные арифметические операции: сложение, вычитание, умножение и деление. Правила выполнения этих операций в десятичной системе хорошо известны — это сложение, вычитание, умножение столбиком и деление углом. Эти правила применимы и ко всем другим позиционным системам счисления. Только нужно пользоваться особыми для каждой системы таблицами сложения и умножения.

Сложение. При сложении цифры суммируются по разрядам; если при этом возникает избыток, то он переносится влево в старший разряд.

Сложение в двоичной системе счисления

+	0	1
0	0	1
1	1	10

Примеры.

Сложение в десятичной системе счисления: $15_{10} + 6_{10}$.

$$\begin{array}{r} 1 \\ + 15 \\ + 6 \\ \hline 21 \end{array}$$

Сложение в двоичной системе счисления: $1111_2 + 110_2$.

$$\begin{array}{r}
 111 \\
 + 1111 \\
 + 110 \\
 \hline
 10101
 \end{array}$$

Рассмотрим еще несколько примеров сложения в двоичной системе счисления:

$$\begin{array}{r}
 1 \\
 + 1001 \\
 \hline
 10011
 \end{array}
 \quad
 \begin{array}{r}
 1111 \\
 + 1101 \\
 + 1011 \\
 \hline
 11000
 \end{array}
 \quad
 \begin{array}{r}
 11111 \\
 + 11111 \\
 + 1 \\
 \hline
 100000
 \end{array}
 \quad
 \begin{array}{r}
 1 \quad 111,1 \\
 + 1010011,111 \\
 + 11001,110 \\
 \hline
 1101101,101
 \end{array}$$

Вычитание. При выполнении операции вычитания всегда из большего по абсолютной величине числа вычитается меньшее число и ставится соответствующий знак. В таблице вычитания точка означает заем в старшем разряде, который переходит в младший разряд как q единиц.

Примеры.

Вычитание в десятичной системе счисления: $201,25_{10} - 59,75_{10}$.

$$\begin{array}{r}
 . . . \\
 201,25 \\
 - 59,75 \\
 \hline
 141,50
 \end{array}$$

Вычитание в двоичной системе счисления: $11001001,01_2 - 111011,11_2$.

$$\begin{array}{r}
 \\
 11001001,01 \\
 - 00111011,11 \\
 \hline
 10001101,10
 \end{array}$$

Рассмотрим еще несколько примеров вычитания в десятичной системе счисления:

$$\begin{array}{r}
 \\
 110110101 \\
 - 101011111 \\
 \hline
 001010110
 \end{array}
 \quad
 \begin{array}{r}
 . . \\
 10111001,1 \\
 - 10001101,1 \\
 \hline
 00101100,0
 \end{array}$$

Умножение. Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необ-

ходимо заимствовать из соответствующих рассматриваемой системе таблиц умножения и сложения.

Умножение в двоичной системе счисления

×	0	1
0	0	0
1	0	1

Примеры.

Умножение в десятичной системе счисления: $115_{10} \cdot 51_{10}$.

$$\begin{array}{r} 115 \\ \times 51 \\ \hline 115 \\ + 575 \\ \hline 5865 \end{array}$$

Умножение $(115 \cdot 51)_{10}$ в двоичной системе счисления: $1110011_2 \cdot 110011_2$.

$$\begin{array}{r} 1110011 \\ 110011 \\ \hline 1110011 \\ + 1110011 \\ 1110011 \\ 1110011 \\ \hline 1011011101001 \end{array}$$

Можно провести проверку, преобразовав полученное произведение в двоичном виде к десятичному:

$$1011011101001_2 = 2^{12} + 2^{10} + 2^9 + 2^7 + 2^6 + 2^5 + 2^3 + 2^0 = 5865.$$

Рассмотрим еще несколько примеров умножения в двоичной системе счисления:

$$\begin{array}{r} \times 1001,1 \\ 10,1 \\ \hline 10011 \\ + 10011 \\ \hline 10111,11 \end{array} \quad \begin{array}{r} \times 1100,01 \\ 11 \\ \hline 110001 \\ + 110001 \\ \hline 100100,11 \end{array} \quad \begin{array}{r} \times 100000,1 \\ 10,01 \\ \hline 1000001 \\ + 1000001 \\ \hline 1001001,001 \end{array}$$

Деление. Деление в любой позиционной системе счисления производится по тем же правилам, что и деление углом в десятичной системе счисления.

тичной системе счисления. В двоичной системе счисления деление выполняется особенно просто: ведь очередная цифра частного может быть только нулем или единицей.

Примеры. Разделить 5865 на 115.

Деление в десятичной системе счисления: $5865_{10} : 115_{10}$.

$$\begin{array}{r} 5865 \overline{)115} \\ - 575 \overline{)51} \\ \hline 115 \\ - 115 \\ \hline 0 \end{array}$$

Деление $(5865 : 115)_{10}$ в двоичной системе счисления: $1011011101001_2 : 1110011_2$.

$$\begin{array}{r} 1011011101001 \overline{)1110011} \\ - 1110011 \overline{)110011} \\ \hline 10001000 \\ - 1110011 \\ \hline 10101100 \\ - 1110011 \\ \hline 1110011 \\ - 1110011 \\ \hline 0 \end{array}$$

Проверка: $110011_2 = 2^5 + 2^4 + 2^1 + 2^0 = 51$.

Рассмотрим еще два примера деления в двоичной системе счисления:

$$75 : 15 = 5$$

$$\begin{array}{r} 1001011 \overline{)1111} \\ - 1111 \overline{)101} \\ \hline 1111 \\ - 1111 \\ \hline 0 \end{array}$$

$$84 : 4 = 21$$

$$\begin{array}{r} 1010100 \overline{)100} \\ - 100 \overline{)10101} \\ \hline 101 \\ - 100 \\ \hline 100 \\ - 100 \\ \hline 0 \end{array}$$

2.1.6. Представление чисел в компьютере

Целые числа могут представляться в компьютере без знака или со знаком.

Целые числа без знака. Целые числа без знака обычно занимают в памяти компьютера 1 или 2 байт и принимают в однобайтовом формате значения от 00000000_2 до 11111111_2 , а в двухбайтовом формате — от $00000000\ 00000000_2$ до $11111111\ 11111111_2$ (табл. 2.2).

Целые числа со знаком. Целые числа со знаком обычно занимают в памяти компьютера 1, 2 или 4 байт, при этом самый левый (старший) разряд содержит информацию о знаке числа. Знак «+» кодируется нулем, а «-» — единицей (табл. 2.3).

Рассмотрим особенности записи целых чисел со знаком на примере однобайтового формата, при котором для знака отводится один разряд, а для цифр абсолютной величины — семь разрядов.

В компьютерной технике применяются три формы записи (кодирования) целых чисел со знаком: прямой код, обратный код, дополнительный код. Последние две формы применяются особенно широко, так как позволяют упростить конструкцию арифметического устройства компьютера путем замены разнообразных арифметических операций операцией сложения.

Положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково — двоичными кодами с цифрой 0 в знаковом разряде.

Таблица 2.2

Диапазоны значений целых чисел без знака

Формат числа, байт	Запись с порядком	Обычная запись
1	$0 \dots 2^8 - 1$	0...255
2	$0 \dots 2^{16} - 1$	0...65535

Таблица 2.3

Диапазоны значений целых чисел со знаком

Формат числа, байт	Запись с порядком	Обычная запись
1	$-2^7 \dots 2^7 - 1$	-128...127
2	$-2^{15} \dots 2^{15} - 1$	-31 768...32 767
4	$-2^{31} \dots 2^{31} - 1$	-2 147 483 648...2 147 483 647

Число $1_{10} = 1_2$:

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

↑ Знак числа «-»

Число $127_{10} = 1111111_2$:

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

↑ Знак числа «-»

Отрицательные числа в прямом, обратном и дополнительном кодах имеют разное изображение.

Прямой код числа -1:

1	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

↑ Знак числа «+»

Прямой код числа -127:

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

↑ Знак числа «+»

Прямой код: в знаковый разряд помещается цифра 1, а в разряды цифровой части числа — двоичный код его абсолютной величины.

Обратный код получается инвертированием всех цифр двоичного кода, абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы — нулями.

Примеры.

Число: -1.

Код модуля числа: 00000001

Обратный код числа: 11111110

Число: -127.

Код модуля числа: 01111111

Обратный код числа: 10000000

Дополнительный код получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду.

Дополнительный код числа -1: Дополнительный код числа -127:

	1	1	1	1	1	1	1	0
+								
							1	
	1	1	1	1	1	1	1	1

	1	0	0	0	0	0	0	0
+								
								1
	1	0	0	0	0	0	0	1

Используя прямой, обратный и дополнительный коды, можно свести операцию умножения к последовательности сложений и сдвигов, а операцию деления к многократному прибавлению к делителю дополнительного кода делителя.

Обычно отрицательные десятичные числа при вводе в машину автоматически преобразуются в обратный или дополнительный двоичный код и в таком виде хранятся, перемещаются и участвуют в операциях. При выводе таких чисел из машины происходит обратное преобразование в отрицательные десятичные числа.

Вещественные числа. Вещественные числа (конечные и бесконечные десятичные дроби) хранятся и обрабатываются в компьютере в формате с плавающей запятой. В этом случае положение запятой в записи числа может изменяться.

Формат чисел с плавающей запятой базируется на экспоненциальной форме записи, в которой может быть представлено любое число. Так, число A может быть представлено в виде

$$A = m \cdot q^n,$$

где m — мантисса числа; q — основание системы счисления; n — порядок числа.

Для однозначности представления чисел с плавающей запятой используется нормализованная форма, при которой мантисса отвечает условию

$$1/n = |m| < 1.$$

Это означает, что мантисса должна быть правильной дробью и иметь после запятой цифру, отличную от нуля.

Примеры. Преобразуем числа в экспоненциальную форму с нормализованной мантиссой:

$$421,637 = 0,421637 \cdot 10^3;$$

$$0,000286 = 0,286 \cdot 10^{-4};$$

$$0,5 = 0,5 \cdot 10^0;$$

$$-25,25 = -2,525 \cdot 10^2.$$

Число в форме с плавающей запятой занимает в памяти компьютера 4 (число обычной точности) или 8 (число двойной точности) байт. При записи числа с плавающей запятой выделяют разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.

Арифметические операции. При сложении и вычитании чисел в формате с плавающей запятой сначала производится подготовительная операция выравнивания порядков. Порядок меньшего (по модулю) числа увеличивается до величины порядка большего (по модулю) числа. Для того чтобы величина числа не изменилась, мантисса уменьшается в такое же число раз (сдвигается в ячейке памяти вправо на число разрядов, равное разности порядков чисел).

После выполнения операции выравнивания одинаковые разряды чисел оказываются расположенными в одних и тех же разрядах ячеек памяти. Теперь операции сложения и вычитания чисел сводятся к сложению или вычитанию мантисс.

После выполнения арифметической операции для приведения полученного числа к стандартному формату с плавающей запятой

производится нормализация, т.е. мантисса сдвигается влево или вправо так, чтобы ее первая значащая цифра попала в первый разряд после запятой.

Пример. Произвести сложение чисел $0,1 \cdot 2^3$ и $0,1 \cdot 2^5$ в формате с плавающей запятой.

Произведем выравнивание порядков и сложение мантисс:

$$\begin{array}{r} 0,001 \cdot 2^5 \\ + 0,100 \cdot 2^5 \\ \hline 0,101 \cdot 2^5 \end{array}$$

При умножении чисел в формате с плавающей запятой порядки складываются, а мантиссы перемножаются. При делении из порядка делимого вычитается порядок делителя, а мантисса делимого делится на мантиссу делителя.

Пример. Произвести умножение чисел $0,1 \cdot 2^3$ и $0,1 \cdot 2^5$ в формате с плавающей запятой.

После умножения будет получено число $0,01 \cdot 2^8$, которое после нормализации примет вид $0,1 \cdot 2^7$.

2.1.7. Представление других видов информации в компьютере

Все виды информации (текстовая, графическая, звуковая, видео-) кодируются в последовательности нулей и единиц. Рассмотрим этот процесс более подробно.

Двоичное кодирование текстовой информации. Традиционно для кодирования одного символа используется количество информации, равное 1 байт, которое составляет 8 бит ($2^8 = 256$), поэтому можно закодировать 256 различных символов.

Такое число символов вполне достаточно для представления текстовой информации, включая прописные и заглавные буквы русского и латинского алфавита, цифры, знаки, графические символы и т.д.

Кодирование заключается в том, что каждому символу ставится в соответствие уникальный десятичный код от 0 до 255 или соответствующий ему двоичный код от 00000000 до 11111111.

Когда пользователь нажимает на клавиатуре клавишу с символом, в память компьютера поступает последовательность из восьми электрических импульсов (двоичный код символа). Код символа хранится в оперативной памяти компьютера, где он занимает одну ячейку памяти.

В процессе вывода символа на экран компьютера производится обратный процесс — декодирование, т.е. преобразование кода символа в его изображение.

Присвоение символу конкретного кода — это вопрос соглашения, которое фиксируется в кодовой таблице. Первые 33 кода (с 0 по 32) соответствуют не символам, а операциям (перевод строки, ввод пробела и т.д.).

В качестве международного стандарта принята кодовая таблица ASCII (American Standard Code for Information Interchange) (рис. 2.1, а), кодирующая первую половину символов с числовыми кодами от 32 до 126.

Хронологически одним из первых стандартов кодирования русских букв на компьютерах был КОИ8 («Код обмена информацией, 8-битный»). Эта кодировка с середины 1980-х гг. стала использоваться в первых русифицированных версиях операционной системы UNIX.

Национальные стандарты кодировочных таблиц включают в себя международную часть кодовой таблицы без изменений и коды национальных алфавитов, символы псевдографики и некоторые математические знаки.

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

а

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

б

Рис. 2.1. Примеры кодировочных таблиц:

а — международная кодировка ASCII; б — кодировка CP1251

Наиболее распространенной в настоящее время является кодировка Microsoft Windows, обозначаемая CP1251 (от *англ.* Code Page — кодовая страница) (рис. 2.1, б).

В конце 1990-х гг. появился новый международный стандарт Unicode, который отводит под один символ не 1 байт, а 2 байт, поэтому с его помощью можно закодировать не 256, а 65 536 различных символов. Полная спецификация стандарта Unicode включает в себя все существующие, вымершие и искусственно созданные алфавиты мира, а также множество математических, музыкальных, химических и прочих символов.

Двоичное кодирование графической информации. Графические изображения, хранящиеся в аналоговой (непрерывной) форме на бумаге, фото- и киноплёнке, могут быть преобразованы в цифровой компьютерный формат путем пространственной дискретизации. Это реализуется путем сканирования, результатом которого является растровое изображение. Растровое изображение состоит из отдельных точек — пикселей (от *англ.* picture element — элемент изображения), каждая из которых может иметь свой цвет.

Качество изображения определяется разрешающей способностью монитора, т. е. числом точек, из которых оно складывается. Чем больше разрешающая способность монитора, т. е. чем больше число строк раstra и точек в строке, тем выше качество изображения. В современных персональных компьютерах обычно используются четыре основные разрешающие способности экрана: 640×480 , 800×600 , 1024×768 и 1280×1024 точки.

Цветные изображения формируются в соответствии с двоичным кодом цвета каждой точки, хранящимся в видеопамяти. Цветные изображения могут иметь различную глубину цвета, которая задается используемым числом бит для кодирования цвета точки (табл. 2.4).

Цветное изображение на экране монитора формируется за счет смешивания трех базовых цветов: красного, зеленого и синего. Такая цветовая модель называется RGB-моделью (от *англ.* Red, Green, Blue — красный, зеленый, синий). Для получения богатой палит-

Таблица 2.4

Глубина цвета и число отображаемых цветов

Глубина цвета	Число отображаемых цветов (N)
4	$2^4 = 16$
8	$2^8 = 256$
16 (High Color)	$2^{16} = 65\,536$
24 (True Color)	$2^{24} = 16\,777\,216$
32 (True Color)	$2^{32} = 4\,294\,967\,296$

Таблица 2.5
Формирование цветов при глубине цвета 24 бит

Цвет	Интенсивность		
	Красный	Зеленый	Синий
Черный	00000000	00000000	00000000
Красный	11111111	00000000	00000000
Зеленый	00000000	11111111	00000000
Синий	00000000	00000000	11111111
Голубой	00000000	11111111	11111111
Желтый	11111111	11111111	00000000
Белый	11111111	11111111	11111111

ры цветов базовым цветам могут быть заданы различные интенсивности. Например, при глубине цвета в 24 бит на каждый из цветов выделяется по 8 бит, т.е. для каждого из цветов возможны $2^8 = 256$ уровней интенсивности, заданных двоичными кодами (от минимальной — 00000000 до максимальной — 11111111) (табл. 2.5).

Для того чтобы на экране монитора формировалось изображение, информация о каждой его точке (код цвета, точки) должна храниться в видеопамяти компьютера. Рассчитаем необходимый объем видеопамяти для одного из графических режимов, например с разрешением 800×600 точек и глубиной цвета 24 бит на точку. Всего точек на экране: $800 \cdot 600 = 480\,000$. Необходимый объем видеопамяти: $24 \text{ бит} \cdot 480\,000 = 11\,520\,000 \text{ бит} = 1\,440\,000 \text{ байт} = 1\,406,25 \text{ Кбайт} = 1,37 \text{ Мбайт}$.

Двоичное кодирование звуковой информации. Звук представляет собой звуковую волну с непрерывно меняющейся амплитудой и частотой. Чем больше амплитуда сигнала, тем он громче для человека; чем больше частота сигнала, тем выше тон. Для того чтобы

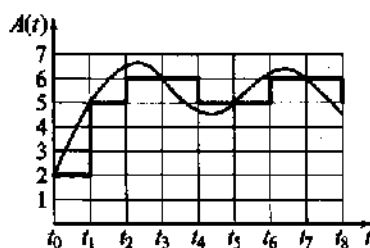


Рис. 2.2. Сетка уровней квантования

компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть превращен в последовательность электрических импульсов (двоичных нулей и единиц).

Таким образом, непрерывная зависимость амплитуды сигнала от времени $A(t)$ заменяется на дискретную последовательность уровней громкости.

Дискретизация — процесс разбиения сигнала на отдельные состав-

ляющие, взятые в определенные тактовые моменты времени $t_0, t_1, t_2, \dots$ через четко определенные тактовые интервалы времени T (рис. 2.2).

Квантование — замена отдельных составляющих исходного дискретного значения сигнала ближайшим уровнем квантования, сдвинутых друг относительно друга на промежуток, называемый шагом квантования:

$$A(t_0) = 2; A(t_1) = 5; A(t_2) = 6; A(t_3) = 6; A(t_4) = 5; A(t_5) = 5; A(t_6) = 6; \\ A(t_7) = 6; A(t_8) = 5.$$

Кодирование — перевод значения уровня квантования в конкретный двоичный код, например:

$$2 - 0010; 6 - 0110; 6 - 0110; \\ 5 - 0101; 5 - 0101; 6 - 0110; \\ 6 - 0110; 5 - 0101; 4 - 0100.$$

Качество передаваемой информации при этом будет зависеть:

- от разрядности преобразования, т.е. числа двоичных разрядов, которые будут использованы при кодировании соответствующего уровня;
- частоты дискретизации — частоты, с которой аналоговый сигнал будет преобразован в цифровую форму с помощью одной из систем счисления.

Уровни громкости звука можно рассматривать как набор возможных состояний. Следовательно, чем большее число уровней громкости будет выделено в процессе кодирования, тем большее количество информации будет нести значение каждого уровня и тем более качественным будет звучание. Звуковые карты обеспечивают, например, 16-битную глубину кодирования звука, обеспечивая $2^{16} = 65\,536$ уровней сигнала.

Кроме того, качество кодирования зависит и от числа точек измерения уровня сигнала за 1 с, т.е. частоты дискретизации (это значение изменяется от 8000 до 48 000).

Принято измерять частоту дискретизации в кГц (килогерцах): 1 кГц — это 1000 измерений в секунду.

Можно оценить информационный объем стереоаудиофайла длительностью звучания 1 с при высоком качестве звука (16 бит, 48 кГц). Для этого число бит на одну выборку необходимо умножить на число выборок в 1 с и умножить на 2 (стереорежим):

$$16 \text{ бит} \cdot 48\,000 \cdot 2 = 1\,536\,000 \text{ бит} = 192\,000 \text{ байт} \approx 187,5 \text{ Кбайт}.$$

Информационный объем звукового файла длительностью 1 мин приблизительно равен 11 Мбайт.

Контрольные вопросы

1. Чем отличаются позиционные системы счисления от непозиционных?
2. Какое количество информации несет в себе цифра восьмеричного числа?
3. Почему в компьютере используется двоичная система счисления?
4. В чем заключается преимущество экспоненциальной формы числа?
5. Как кодируются символы текста?
6. В чем заключается метод пространственной дискретизации?
7. Переведите в десятичную систему счисления 1110_2 ; 22_8 ; BF_{16} ; 10110_2 ; 135_8 ; $70E_{16}$.
8. Переведите десятичные числа в двоичную, восьмеричную и шестнадцатеричную системы счисления: 74,21; 26,11; 125,01; 114,08.
9. Переведите пары чисел в двоичную систему счисления, произведите арифметические операции, ответы проверьте: 36 и 4; 75 и 5; 12 и 4; 123 и 3.

10. В какой системе счисления справедливы следующие равенства:

$$20 + 25 = 100; 22 + 44 = 110?$$

11. Десятичное число 59 эквивалентно числу 214 в некоторой другой системе счисления. Найдите основание этой системы.

12. Переведите числа в десятичную систему, а затем проверьте результаты, выполнив обратные переводы:

1011011_2	10110111_2	011100001_2
$0,1000110_2$	$110100,11_2$	517_8
1010_8	1234_8	$0,34_8$
$123,41_8$	$1F_{16}$	ABC_{16}
1010_{16}	$0,44_{16}$	$1DE, C8_{16}$

13. Вычитите:

$$\begin{array}{ll} 10100_2 - 111_2 & 100,1_2 - 10,11_2 \\ 10010_2 - 111,1_2 & 1110,11_2 - 10001_2 \end{array}$$

14. Перемножьте числа, а затем проверьте результаты, выполнив соответствующие десятичные умножения:

$$\begin{array}{ll} 101101_2 \cdot 101_2 & 111101_2 \cdot 11,01_2 \\ 1011,11_2 \cdot 101,1_2 & 101_2 \cdot 1111,001_2 \end{array}$$

15. Разделите 10010110_2 на 1010_2 и проверьте результат, умножив делитель на частное.

16. Запишите числа в прямом коде (формат 1 байт):

$$\begin{array}{ll} 31 & -63 \\ 65 & -128 \end{array}$$

17. Запишите числа в обратном и дополнительном кодах (формат 1 байт):

$$\begin{array}{ll} -9 & -15 \\ -127 & -128 \end{array}$$

18. Найдите десятичные представления чисел, записанных в дополнительном коде:

11111000	10011011
11101001	10000000

19. Выполните вычитания чисел путем сложения их обратных (дополнительных) кодов в формате 1 байт (укажите, в каких случаях имеет место переполнение разрядной сетки):

9 – 2	2 – 9
–5 – 7	–20 – 10
50 – 25	127 – 1

20. Закодируйте с помощью таблицы CP1251 и представьте в шестнадцатеричной системе счисления слово «информация».

21. Почему иногда на экране монитора вместо текстовой информации можно видеть 0000 и т. п.?

22. На клавиатуре наряду с алфавитно-цифровыми клавишами размещены такие, как [Alt], [Ctrl], [Del] и т. д. Имеют ли они десятичный код?

23. При разрешающей способности 1280×1024 точек определите объем видеопамати при глубине цвета High Color.

24. Сколько может «весить», т. е. какой имеет объем, файл с видеоклипком длительностью 5 с?

25. Сколько точек содержит рисунок, если при кодировании каждой точки 1 байт получился файл объемом 300 Кбайт?

2.2. ЛОГИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ КОМПЬЮТЕРА

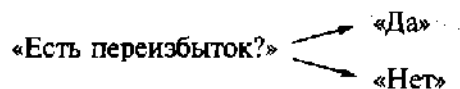
Из подразд. 2.1 вы узнали, что любая информация в компьютере (или в другом устройстве вычислительной техники) представлена в виде двоичного цифрового сигнала, что правила выполнения операций в двоичной системе достаточно просты.

Выполним фрагмент алгоритма перевода целого десятичного числа в двоичную систему счисления: «Последовательно выполнять деление числа и получаемых целых частных на 2 до тех пор, пока не получим частное меньше 2».

Компьютер, который автоматически выполняет операции перевода, должен уметь делать выбор:

«Полученное целое частное меньше 2?»	→ «Да» — закончить работу
	→ «Нет» — продолжить деление

При выполнении двоичного сложения необходимо придерживаться следующего правила: «Цифры суммируются по разрядам, и если при этом возникает избыток, то единица переносится влево в старший разряд», т. е. постоянно нужно анализировать:



2.2.1. Алгебра логики

Алгебра логики появилась в середине XIX в. в трудах английского математика Джорджа Буля. Он пытался решать традиционные логические задачи алгебраическими методами.

Логическое высказывание — это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно. Например, предложение «6 — четное число» следует считать высказыванием, так как оно истинное; предложение «Рим — столица Франции» — тоже высказывание, так как оно ложное.

Разумеется, не всякое предложение является логическим высказыванием.

Высказываниями не являются, например, предложения: «Студент 204 группы», «Здравствуйте!», «В колледже более 1000 учащихся», «Хороший студент», — так как невозможно судить об их истинности или ложности (или нужны дополнительные сведения, чтобы предложение стало высказыванием, например: «Петров — хороший студент» или «в колледже № 15 г. Самары более 1000 учащихся»).

Алгебра логики рассматривает любое высказывание только с одной точки зрения — является ли оно истинным или ложным.

Заметим, что зачастую трудно установить истинность высказывания. Например, высказывание «Площадь поверхности Индийского океана равна 75 млн км²» в одной ситуации можно считать ложным, а в другой — истинным (ложным — так как указанное значение неточное и вообще не является постоянным; истинным — если рассматривать его как некоторое приближение, приемлемое на практике).

Употребляемые в обычной речи слова и словосочетания «не», «и», «или», «если, то», «тогда и только тогда» и другие позволяют из уже заданных высказываний строить новые. Такие слова и словосочетания называются *логическими связками*.

Высказывания, образованные из других высказываний с помощью логических связок, называются *составными*. Высказывания, не являющиеся составными, называются *элементарными*. Например, из элементарных высказываний «Иванов — студент», «Иванов — отличник» при помощи связки И можно получить составное высказывание «Иванов — студент и отличник».

Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через А обозначено высказывание «Иван поедет летом на море», а через В — высказывание «Иван летом отправит-

ся в горы». Тогда составное высказывание «Иван летом побывает и на море, и в горах» можно записать кратко: «А И В». Здесь И — логическая связка; А, В — логические переменные, которые могут принимать только два значения: «истина» и «ложь», обозначаемые соответственно 1 и 0.

В алгебре высказывания обозначаются именами логических переменных, которые могут принимать лишь два значения: «истина» (1) и «ложь» (0).

Рассмотрим операции, которые можно производить с логическими высказываниями.

Операция отрицания. Операция, выражаемая словом НЕ, называется отрицанием, или инверсией, и обозначается чертой над высказыванием. Высказывание А истинно, когда А ложно, и ложно, когда А истинно. Например: «Луна — спутник Земли» (А); «Луна — не спутник Земли» (А).

Пусть А = «Два умножить на два равно четырем» — истинное высказывание, тогда высказывание, образованное с помощью операции логического отрицания — «Два умножить на два не равно четырем», — ложное высказывание.

Образует высказывание F, являющееся логическим отрицанием А:

$$F = \bar{A}.$$

Истинность такого высказывания задается таблицей истинности функции логического отрицания (табл. 2.6).

Таблица 2.6

Таблица истинности функции логического отрицания

A	F
0	1
1	0

Операция конъюнкции. Операция, выражаемая связкой И, называется соединением, или конъюнкцией, или логическим умножением, и обозначается знаком «&» (может обозначаться знаком «^» или «·»). Высказывание $F = A \& B$ истинно только тогда, когда оба высказывания А и В истинны. Например, высказывание «10 делится на 2 И 5 больше 3» истинно, а высказывания «10 делится на 2 И 5 не больше 3», «10 не делится на 2 И 5 больше 3», «10 не делится на 2 И 5 не больше 3» ложны.

Значение логической функции F можно определить с помощью таблицы истинности данной функции, которая показывает, какие значения принимает логическая функция при всех возможных наборах ее аргументов (табл. 2.7).

Таблица 2.7

Таблица истинности функции логического умножения

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Рассмотрим, например, составное высказывание « $2 \cdot 2 = 4$ И $3 \cdot 3 = 10$ ». Первое простое высказывание истинно ($A = 1$), а второе высказывание ложно ($B = 0$). По табл. 2.7 определяем, что логическая функция принимает значение «ложь» ($F = 0$), т.е. данное составное высказывание ложно.

Операция дизъюнкции. Операция, выражаемая связкой ИЛИ, называется разделением, или дизъюнкцией (от лат. *disjunctio* — разделение), или логическим сложением, и обозначается знаком « $\vee$ » (или «+»). Высказывание $F = A \vee B$ ложно тогда и только тогда, когда оба высказывания A и B ложны. Например, высказывание «10 не делится на 2 ИЛИ 5 не больше 3» ложно, а высказывания «10 делится на 2 ИЛИ 5 больше 3», «10 делится на 2 ИЛИ 5 не больше 3», «10 не делится на 2 ИЛИ 5 больше 3» истинны.

Функцию F можно определить с помощью таблицы истинности, которая показывает, какие значения принимает логическая функция при всех возможных наборах ее аргументов (табл. 2.8).

Таблица 2.8

Таблица истинности функции логического сложения

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Операция импликации. Операция следования, выражаемая связками «если..., то», «из ... следует», «... влечет...», называется импликацией и обозначается знаком « $\rightarrow$ ». Высказывание $A \rightarrow B$ ложно тогда и только тогда, когда A истинно, а B ложно.

Каким же образом импликация связывает два элементарных высказывания? Покажем это на примере следующих высказываний: «Данный четырехугольник — квадрат» (A) и «Около данного четырехугольника можно описать окружность» (B). Рассмотрим составное высказывание $F = A \rightarrow B$, под которым понимается: «Если данный четырехугольник — квадрат, то около него можно описать окружность».

Существуют три варианта, когда высказывание $A \rightarrow B$ истинно:

- 1) A истинно и B истинно, т.е. данный четырехугольник — квадрат и около него можно описать окружность;
- 2) A ложно и B истинно, т.е. данный четырехугольник не является квадратом, но около него можно описать окружность (разумеется, это справедливо не для всякого четырехугольника);
- 3) A ложно и B ложно, т.е. данный четырехугольник не является квадратом и около него нельзя описать окружность.

Ложен только один вариант: A истинно и B ложно, т.е. данный четырехугольник является квадратом, но около него нельзя описать окружность.

Функцию F можно определить, используя таблицу истинности (табл. 2.9): $F = A \rightarrow B$

Таблица 2.9

Таблица истинности логической функции импликации

A	B	F
0	0	1
0	1	1
1	0	0
1	1	1

Операция логического следования несколько отличается от обычного понимания слова «следует». Если первое высказывание (предпосылка) ложно, то независимо от истинности или ложности второго высказывания (вывода) составное высказывание истинно. Из неверной предпосылки может следовать что угодно.

В алгебре высказываний все логические функции могут быть сведены путем логических преобразований к трем базовым: логическому умножению, логическому сложению и логическому отрицанию.

Докажем методом сравнения таблиц истинности, что операция импликации $A \rightarrow B$ равносильна логическому выражению $A \vee \bar{B}$ (табл. 2.10).

Таблица 2.10

Таблица истинности логического выражения $A \vee \bar{B}$

A	B	$\bar{A}$	$A \vee \bar{B}$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Табл. 2.10 полностью совпадает с табл. 2.9.

Операция эквиваленции. Операция равенства, выражаемая связками «тогда и только тогда», «необходимо и достаточно», «... равносильно...», называется эквиваленцией, или двойной импликацией, и обозначается знаками « $\leftrightarrow$ » или « $\sim$ ». Высказывание $A \leftrightarrow B$ истинно тогда и только тогда, когда значения A и B совпадают. Например, истинны высказывания: «24 делится на 6 тогда и только тогда, когда 24 делится на 3», «23 делится на 6 тогда и только тогда, когда 23 делится на 3» — и ложны высказывания: «24 делится на 6 тогда и только тогда, когда 24 делится на 5», «21 делится на 6 тогда и только тогда, когда 21 делится на 3».

Высказывания A и B , образующие составное высказывание $F = A \leftrightarrow B$, могут быть совершенно не связаны по содержанию.

Составное высказывание $F = A \leftrightarrow B$ имеет таблицу истинности (табл. 2.11).

Таблица 2.11

Таблица истинности составного высказывания $A \leftrightarrow B$

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Составное высказывание, образованное с помощью логической операции эквивалентности, истинно тогда и только тогда, когда оба высказывания одновременно либо ложны, либо истинны.

Эквивалентность можно выразить через следующие логические функции:

$$F = A \leftrightarrow B = (\bar{A} \vee B) \& (\bar{B} \vee A).$$

Порядок выполнения логических операций задается круглыми скобками. Но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания (НЕ), затем — конъюнкции (И), затем дизъюнкции (ИЛИ) и в последнюю очередь — импликации. Такая последовательность называется приоритетом операций.

2.2.2. Основные законы алгебры логики

В алгебре логики имеется ряд законов, позволяющих производить равносильные преобразования логических выражений.

1. Закон двойного отрицания:

$$A = \overline{\overline{A}}.$$

Двойное отрицание исключает отрицание.

2. Переместительный (коммутативный) закон:

- для логического сложения

$$A \vee B = B \vee A;$$

- для логического умножения

$$A \& B = B \& A.$$

Результат операции над высказываниями не зависит от того, в каком порядке берутся эти высказывания.

В обычной алгебре $a + b = b + a$; $a \cdot b = b \cdot a$.

3. Сочетательный (ассоциативный) закон:

- для логического сложения

$$(A \vee B) \vee C = A \vee (B \vee C);$$

- для логического умножения

$$(A \& B) \& C = A \& (B \& C).$$

При одинаковых знаках скобки можно ставить произвольно или опускать.

В обычной алгебре $(a + b) + c = a + (b + c) = a + b + c$;

$$(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c.$$

4. Распределительный (дистрибутивный) закон:

- для логического сложения

$$(A \vee B) \& C = (A \& C) \vee (B \& C);$$

- для логического умножения

$$(A \& B) \vee C = (A \vee C) \& (B \vee C).$$

Данный закон определяет правило выноса общего высказывания за скобку.

В обычной алгебре справедлив распределительный закон только для сложения: $(a + b) \cdot c = a \cdot c + b \cdot c$.

5. Закон общей инверсии (законы де Моргана):

- для логического сложения

$$\overline{A \vee B} = \overline{A} \& \overline{B};$$

- для логического умножения

$$\overline{A \& B} = \overline{A} \vee \overline{B}.$$

6. Закон идемпотентности (от *лат. idem* — тот же самый + *potens* — сильный (дословно — равносильный)):

- для логического сложения

$$A \vee A = A$$

- для логического умножения

$$A \& A = A.$$

Закон означает отсутствие показателей степени.

7. Закон исключения констант:

- для логического сложения

$$A \vee 1 = 1; A \vee 0 = A;$$

- для логического умножения

$$A \& 1 = A; A \& 0 = 0.$$

8. Закон противоречия:

$$A \& \bar{A} = 0.$$

Невозможно, чтобы противоречащие высказывания были одновременно истинными.

9. Закон исключения третьего:

$$A \vee \bar{A} = 1.$$

Из двух противоречащих высказываний об одном и том же предмете одно всегда истинно, а второе — ложно; третьего не дано.

10. Закон поглощения:

- для логического сложения:

$$A \vee (A \& B) = A;$$

- для логического умножения

$$A \& (A \vee B) = A.$$

11. Закон исключения (склеивания):

- для логического сложения

$$(A \& B) \vee (\bar{A} \& B) = B;$$

- для логического умножения

$$(A \vee B) \& (\bar{A} \vee B) = B.$$

12. Закон контрапозиции (правило перевертывания):

$$(A \leftrightarrow B) = (B \leftrightarrow A).$$

Справедливость приведенных законов можно доказать табличным способом: выписать все наборы значений A и B , вычислить на них значения левой и правой частей доказываемого выражения и убедиться, что значения в результирующих столбцах совпадут.

Пример 1. Найдите X , если $\overline{X \vee A} \vee \overline{X \vee A} = B$.

Для преобразования левой части равенства последовательно воспользуемся законом де Моргана для логического сложения и законом двойного отрицания:

$$(\overline{X \vee A}) \vee (\overline{X \vee A}).$$

Согласно распределительному закону для логического сложения

$$X \& (A \vee A).$$

Согласно закону исключения третьего и закона исключения констант

$$\overline{X \& 1} = \overline{X}.$$

Полученную левую часть приравняем правой:

$$\overline{X} = B.$$

Окончательно получим: $X = \overline{B}$.

Пример 2. Упростите логическое выражение $(A \vee B \vee C) \& \overline{A \vee B \vee C}$. Правильность упрощения проверьте с помощью таблиц истинности для исходного и полученного логических выражений.

Согласно закону общей инверсии для логического сложения (первому закону де Моргана) и закону двойного отрицания

$$(A \vee B \vee C) \& \overline{A \vee B \vee C} = (A \vee B \vee C) \& (\overline{A} \& \overline{B} \& \overline{C}).$$

Согласно распределительному (дистрибутивному) закону для логического сложения

$$(A \vee B \vee C) \& (\overline{A} \& \overline{B} \& \overline{C}) = (A \& \overline{A}) \vee (B \& \overline{A}) \vee (C \& \overline{A}) \vee (A \& \overline{B}) \vee (B \& \overline{B}) \vee (C \& \overline{B}) \vee (A \& \overline{C}) \vee (B \& \overline{C}) \vee (C \& \overline{C}).$$

Согласно закону противоречия

$$(A \& \overline{A}) = 0; (C \& \overline{C}) = 0.$$

Согласно закону идемпотентности

$$(B \& B) = B.$$

Подставляем значения и, используя переместительный (коммутативный) закон и группируя слагаемые, получаем:

$$0 \vee (A \& B) \vee (\bar{A} \& B) \vee B \vee (C \& B) \vee (\bar{C} \& B) \vee (C \& \bar{A}) \vee (A \& \bar{C}) \vee 0.$$

Согласно закону исключения (склеивания)

$$(A \& B) \vee (\bar{A} \& B) = B; (C \& B) \vee (\bar{C} \& B) = B.$$

Подставляем значения и получаем:

$$0 \vee B \vee B \vee B \vee (C \& \bar{A}) \vee (A \& \bar{C}) \vee 0.$$

Согласно закону исключения констант для логического сложения и закону идемпотентности

$$0 \vee B \vee 0 \vee B \vee B = B.$$

Подставляем значения и получаем:

$$B \vee (C \& \bar{A}) \vee (A \& \bar{C}).$$

Согласно распределительному (дистрибутивному) закону для логического умножения

$$(C \& \bar{A}) \vee (A \& C) = (C \vee A) \& (C \vee C) \& (\bar{A} \vee A) \& (A \vee C).$$

Согласно закону исключения третьего

$$(C \vee \bar{C}) = 1; (A \vee A) = 1.$$

Подставляем значения и окончательно получаем:

$$B \& \bar{A} \& \bar{C}.$$

Решение логических задач способствует развитию абстрактного мышления, тренирует память и развивает логику.

2.2.3. Логические основы устройства компьютера

Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: 1 (true) и 0 (false).

Из этого следует два вывода:

- одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных.

- на этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьютера, и, следовательно, уменьшить число элементарных логических элементов, из десятков тысяч которых состоят основные узлы компьютера.

Логический элемент компьютера — это часть электронной логической схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ и др. (называемые также вентилями), а также триггер. С помощью этих схем можно реализовать любую логическую функцию, описывающую работу устройств компьютера. Обычно у вентилях бывает от двух до восьми входов и один или два выхода. Чтобы представить два логических состояния 1 и 0 в вентилях, соответствующие им входные и выходные сигналы имеют один из двух установленных уровней напряжения. Например: 5 и 0 В.

Высокий уровень обычно соответствует значению «истина» (1), а низкий — значению «ложь» (0).

Каждый логический элемент имеет свое условное обозначение, которое выражает его логическую функцию, но не указывает на то, какая именно электронная схема в нем реализована. Это упрощает запись и понимание сложных логических схем.

Работу логических элементов описывают с помощью таблиц истинности (аналогично таблицам истинности функций).

Рассмотрим структурные схемы логических элементов компьютера и их таблицы истинности (табл. 2.12).

- схема И реализует конъюнкцию двух или более логических значений. На выходе схемы И будет 1 тогда и только тогда, когда на всех входах будут 1. Когда хотя бы на одном входе будет 0, на выходе также будет 0;

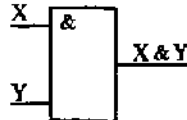
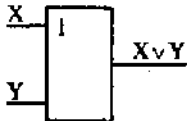
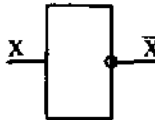
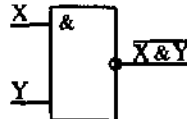
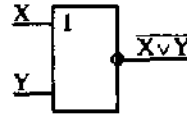
- схема ИЛИ реализует дизъюнкцию двух или более логических значений. Когда хотя бы на одном из выходов схемы или будет 1, на ее выходе тоже будет 1;

- схема НЕ (инвертор) реализует операцию отрицания. Связь между входом x этой схемы и выходом z можно записать соотношением $z = \bar{x}$. Если на входе схемы 0, то на выходе 1. Когда на входе 1, то на выходе 0;

- схема И—НЕ состоит из элемента И и инвертора и осуществляет отрицание результата схемы И. Связь между выходом z и входами x и y схемы записывают следующим образом: $z = x \& y$;

Таблица 2.12

**Структурные схемы логических элементов компьютера
и их таблицы истинности**

Условное обозначение	Структурная схема	Таблица истинности															
И		<table><tr><td>x</td><td>y</td><td>x&y</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	x&y	0	0	0	0	1	0	1	0	0	1	1	1
x	y	x&y															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
ИЛИ		<table><tr><td>x</td><td>y</td><td>x∨y</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	x∨y	0	0	0	0	1	1	1	0	1	1	1	1
x	y	x∨y															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
НЕ		<table><tr><td>x</td><td>$\bar{x}$</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	$\bar{x}$	0	1	1	0									
x	$\bar{x}$																
0	1																
1	0																
И–НЕ		<table><tr><td>x</td><td>y</td><td>x&$\bar{y}$</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	x& $\bar{y}$	0	0	1	0	1	1	1	0	1	1	1	0
x	y	x& $\bar{y}$															
0	0	1															
0	1	1															
1	0	1															
1	1	0															
ИЛИ–НЕ		<table><tr><td>x</td><td>y</td><td>x∨$\bar{y}$</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	x∨ $\bar{y}$	0	0	1	0	1	0	1	0	0	1	1	0
x	y	x∨ $\bar{y}$															
0	0	1															
0	1	0															
1	0	0															
1	1	0															

• схема ИЛИ–НЕ состоит из элемента ИЛИ и инвертора и осуществляет отрицание результата схемы ИЛИ. Связь между выходом z и входами x и y схемы записывают следующим образом:
 $z = x \vee y$.

2.2.4. Триггер

Важнейшей структурной единицей оперативной памяти компьютера, а также внутренних регистров процессора, является триггер. Это устройство позволяет запоминать, хранить и считывать информацию (каждый триггер может хранить 1 бит информации).

Триггер можно построить из двух логических элементов ИЛИ и двух элементов НЕ (рис. 2.3, *а*). Им соответствует таблица истинности (табл. 2.13).

Самый распространенный тип триггера — RS-триггер (от *англ.* set — установка + reset — сброс). Он имеет два симметричных входа R и S и два симметричных выхода $\bar{Q}$ и Q . На каждый из двух выходов могут подаваться входные сигналы в виде импульсов (наличие импульса на входе будем считать единицей, а его отсутствие — нулем). В обычном состоянии на входы R и S триггера подан сигнал 0 и триггер хранит 0. Для записи 1 на вход S (установочный) подается сигнал 1. Последовательно рассмотрев прохождение сигнала по схеме, видно, что триггер переходит в состоя-

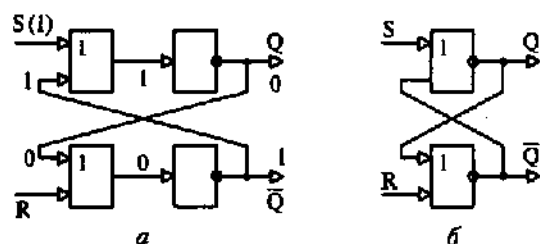


Рис. 2.3. Схема триггера:

а — на элементах ИЛИ и НЕ; *б* — на элементах ИЛИ-НЕ

Таблица 2.13

Таблица истинности

S	R	Q	$\bar{Q}$
0	0	Запрещено	
0	1	1	0
1	0	0	1
1	1	Хранение бита	

ние «1» и будет устойчиво находиться в нем и после того, как сигнал на входе S исчезнет. Триггер запомнил 1, т.е. с выхода триггера Q можно считать 1.

Для того чтобы сбросить информацию и подготовиться к приему новой, подается сигнал 1 на вход R (сброс), после чего триггер возвращается к исходному (нулевому) состоянию. Если на входы R и S подана логическая 1, то состояние $\bar{Q}$ и Q не меняется, подача на оба входа логического 0 может привести к неординарному результату, и поэтому эта комбинация входных сигналов запрещена.

На рис. 2.3, б показана реализация триггера с помощью вентиля ИЛИ—НЕ.

2.2.5. Сумматор двоичных чисел

В целях максимального упрощения работы компьютера все многообразие математических операций в процессоре сводится к сложению двоичных чисел, поэтому важным элементом процессора является сумматор, который и обеспечивает такое сложение.

При сложении двоичных чисел образуется сумма в текущем разряде; при этом возможен перенос единицы в старший разряд. Обозначим слагаемые — A, B, перенос — P и сумму — S.

Таблица 2.14

Таблица сложения одноразрядных двоичных чисел

Слагаемые		Перенос	Сумма
A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Из табл. 2.14 видно, что перенос единицы можно реализовать с помощью операции логического умножения:

$$P = A \& B,$$

где P — перенос; A и B — множители.

Для определения суммы можно применить следующее выражение:

$$S = (A \vee B) \& (\overline{A \& B}).$$

Построим таблицу истинности для данного логического выражения и убедимся в правильности нашего предположения (табл. 2.15).

Таблица 2.15

Таблица истинности функции $F = (A \vee B) \& (\overline{A \& B})$

A	B	$A \vee B$	$A \& B$	$\overline{A \& B}$	$(A \vee B) \& (\overline{A \& B})$
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

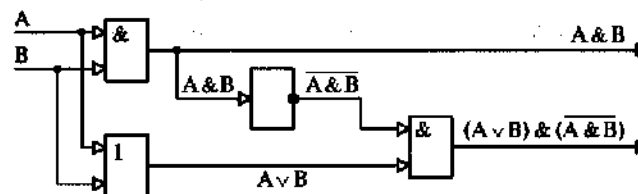


Рис. 2.4. Схема полусумматора двоичных чисел

На основе полученных логических выражений можно построить схему полусумматора на базовых логических элементах.

По логической формуле переноса легко определить, что для получения переноса необходимо использовать логический элемент И.

Анализ логической формулы для суммы показывает, что на выходе должен стоять элемент логического умножения И, который имеет два входа. На один из входов подается результат логического сложения исходных величин $A \vee B$, т.е. на него должен подаваться сигнал с элемента логического сложения ИЛИ.

На второй вход требуется подать результат инвертированного логического умножения исходных сигналов $A \& B$, т.е. на второй вход подается сигнал с элемента НЕ, на вход которого поступает сигнал с элемента логического умножения И (рис. 2.4).

Данная схема называется полусумматором, так как реализует суммирование одноразрядных двоичных чисел без учета переноса из младшего разряда.

Контрольные вопросы

1. Связано ли появление алгебры логики с разработкой персонального компьютера?
2. Назовите основные логические операции.
3. Приведите примеры предложений, которые не являются логическим высказыванием.

4. Покажите связь между алгеброй логики и двоичным кодированием информации.
5. Какой логический элемент нужно поставить в старший разряд, чтобы запомнить целое отрицательное число -5 ?
6. Назовите приоритеты логических операций.
7. Сформулируйте отрицание следующих высказываний: « $2 \geq 5$ »; « $10 < 7$ »; « $a = 2$ ».
8. Изобразите в декартовой системе координат области ($|x| \leq 1$) и ($|y| \leq 1$).
9. Определите истинность составного высказывания « $(\bar{A} \& \bar{B}) \& (C \vee D)$ », состоящего из простых высказываний: A = «Принтер — устройство вывода информации», B = «Процессор — устройство хранения информации», C = «Монитор — устройство вывода информации», D = «Клавиатура — устройство обработки информации».
10. Докажите, используя таблицы истинности, что операция эквивалентности равносильна логическим выражениям $A \leftrightarrow B = (\bar{A} \& \bar{B}) \vee (A \& B)$ и $A \leftrightarrow B = (A \vee \bar{B}) \& (\bar{A} \vee B)$.
11. Докажите, используя таблицы истинности, что логические выражения $\bar{A} \vee \bar{B}$ и $A \& B$ равносильны.
12. Какие логические функции двух аргументов имеют свои названия?
13. Какое существует число логических функций трех аргументов?
14. Упростите следующие логические выражения:

$$(A \vee \bar{A}) \& B;$$

$$A \& (A \vee B) \& (B \vee \bar{B}).$$

15. Приведите примеры из повседневной жизни:
если (не a и не b), то (c или d);
(a или b) тогда и только тогда, когда (c или не d).
16. Проследите на логической схеме триггера, что происходит при поступлении сигнала 1 на вход R .
17. Какое число базовых логических элементов необходимо для хранения 512 Мбайт информации?

2.3. ПРОГРАММНОЕ УПРАВЛЕНИЕ КОМПЬЮТЕРОМ

Первая вычислительная машина, способная хранить программу в своей памяти, разрабатывалась в 1943—1948 гг. в США под руководством Джона Мочли и Преснера Экерта.

В 1945 г. к работе был привлечен знаменитый математик Джон фон Нейман, который сформулировал общие принципы функционирования универсальных вычислительных устройств.

Первый компьютер, в котором были полностью реализованы эти принципы, был построен в 1949 г. английским исследователем Морисом Уилксом. Изменяется элементная база, компьюте-

ры становятся все более и более мощными, но до сих пор большинство из них соответствуют тем принципам, которые изложил в своем докладе в 1945 г. Джон фон Нейман.

2.3.1. Принципы фон Неймана

Устройства компьютера. В состав компьютера должны входить следующие устройства:

- арифметико-логическое устройство, выполняющее арифметические и логические операции;
- устройство управления, которое организует процесс выполнения программ;
- запоминающее устройство, или память, для хранения программ и данных;
- внешние устройства для ввода-вывода информации.

В современных компьютерах это:

- память (запоминающее устройство — ЗУ), состоящая из перенумерованных ячеек;
- процессор, включающий в себя устройство управления (УУ) и арифметико-логическое устройство (АЛУ);
- устройство ввода;
- устройство вывода.

Эти устройства соединены между собой каналами связи, по которым передается информация. Основные устройства компьютера и связи между ними представлены на рис. 2.5.

Функции памяти:

- прием информации из других устройств;

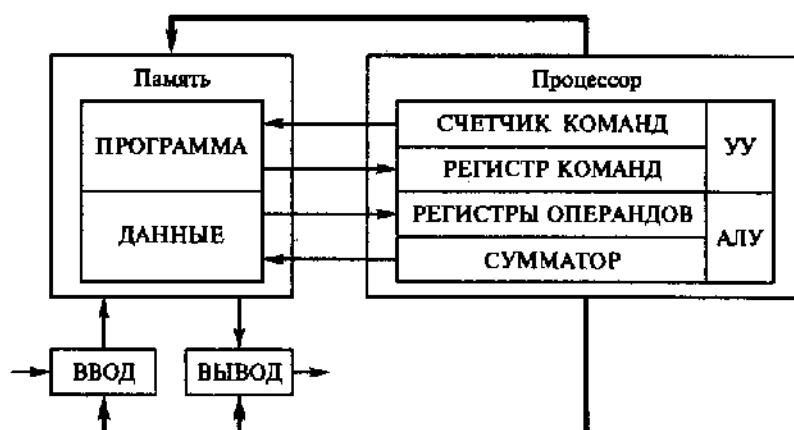


Рис. 2.5. Общая схема компьютера:

— сигнал управления; — информационный сигнал

- запоминание информации;
 - выдача информации по запросу в другие устройства машины.
- Функции процессора:
- обработка данных по заданной программе путем выполнения арифметических и логических операций;
 - программное управление работой устройств компьютера.

Одна часть процессора, которая выполняет команды, называется арифметико-логическим устройством, а другая его часть, выполняющая функции управления устройствами, — устройством управления. Обычно эти устройства выделяются чисто условно, конструктивно они не разделены.

В составе процессора имеется ряд специализированных дополнительных ячеек памяти, называемых регистрами. Регистр выполняет функцию кратковременного хранения числа или команды. Основным элементом регистра является электронная схема, называемая триггером (см. подразд. 2.2).

Регистр представляет собой совокупность триггеров, связанных друг с другом определенным образом общей системой управления.

Существует несколько типов регистров, отличающихся видом выполняемых операций. Некоторые важные регистры имеют свои названия, например:

- сумматор — регистр АЛУ, участвующий в выполнении каждой операции (см. подразд. 2.2);
- счетчик команд — регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды. Он служит для автоматической выборки программы из последовательных ячеек памяти;
- регистр команд — регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения. Часть его разрядов используется для хранения кода операции, остальные — для хранения кодов адресов операндов.

Принцип программного управления. Программа состоит из набора команд, выполняющихся процессором автоматически в определенной последовательности.

Вначале с помощью какого-либо внешнего устройства в память компьютера вводится программа. Устройство управления считывает содержимое ячейки памяти, где находится первая инструкция (команда) программы, и организует ее выполнение. Эта команда может задавать выполнение арифметических или логических действий, чтение из памяти данных для выполнения арифметических или логических операций или запись их результатов в память, ввод данных из внешнего устройства в память или вывод данных из памяти на внешнее устройство.

Выборка программы из памяти осуществляется с помощью счетчика команд. Этот регистр процессора последовательно увеличивает хранимый в нем адрес очередной команды на длину команды. Так как команды расположены в памяти друг за другом,

организуется выборка цепочки команд из последовательно расположенных ячеек памяти. Если же нужно после выполнения команды перейти не к следующей, а к какой-то другой ячейке, то используются команды условного и безусловного перехода, которые заносят в счетчик команд номер ячейки памяти, содержащей следующую команду. Выборка команд из памяти прекращается после достижения и выполнения команды «стоп». Процессор исполняет программу автоматически, без вмешательства человека.

Принцип однородности памяти. Программы и данные хранятся в одной и той же памяти, поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Это открывает целый ряд возможностей. Например, программа в процессе своего выполнения также может подвергаться переработке, что позволяет задавать в самой программе правила получения повтора некоторых ее частей (так в программе организуется выполнение циклов и подпрограмм).

Более того, команды одной программы могут быть получены как результаты выполнения другой программы. На этом принципе основаны методы трансляции — перевода текста программы с языка программирования высокого уровня на язык конкретной машины.

Принцип адресности. Структурно основная память состоит из пронумерованных ячеек. Процессору в произвольный момент времени доступна любая ячейка.

Память компьютера должна состоять из некоторого числа пронумерованных ячеек, в каждой из которых могут находиться или обрабатываемые данные, или инструкции программ. Все ячейки памяти должны быть одинаково легко доступны для других устройств компьютера.

Отсюда следует возможность давать имена областям памяти так, чтобы к сохраненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

2.3.2. Структура и виды команд

Алгоритм решения задачи, заданный в виде последовательности команд на языке вычислительной машины (в кодах машины), называется *машинной программой*.

Команда машинной программы (машинная команда) — это элементарная инструкция машине, выполняемая ею автоматически без каких-либо дополнительных указаний и пояснений.

Команда — это описание элементарной операции, которую должен выполнить компьютер.

Машинная команда состоит из двух частей: операционной и адресной.

Операционная часть команды — это группа разрядов в команде, предназначенная для представления кода операции машины.

Адресная часть команды — это группа разрядов в команде, в которых записываются коды адреса (адресов) ячеек памяти машины, предназначенных для оперативного хранения информации; или иных объектов, задействованных при выполнении команды. Часто эти адреса называют адресами операндов, т. е. чисел, участвующих в операции.

По числу адресов, записываемых в команде, команды подразделяются на трех-, двух-, одноадресные и безадресные (рис. 2.6).

В трехадресной команде:

a1 и a2 — адреса ячеек (регистров), где расположены соответственно первое и второе числа, участвующие в операции;

a3 — адрес ячейки (регистра), куда следует поместить число, полученное в результате выполнения операции.

В двухадресной команде:

a1 — это обычно адрес ячейки (регистра), где хранится первое из чисел, участвующих в операции, и куда после завершения операции должен быть записан результат операции;

a2 — обычно адрес ячейки (регистра), где хранится второе участвующее в операции число.

В одноадресной команде:

a1 — в зависимости от модификатора команды может обозначать либо адрес ячейки (регистра), где хранится одно из чисел, участвующих в операции, либо адрес ячейки (регистра), куда следует поместить число — результат операции.

Безадресная команда содержит только код операции, а информация для нее должна быть заранее помещена в определенные регистры машины (безадресные команды могут использоваться только совместно с командами другой адресности).

Команды хранятся в ячейках памяти в двоичном коде.

Выполнение команды разбивается на следующие этапы:



Рис. 2.6. Типовые структуры команд:

КОП — код операции

- 1) из ячейки памяти, адрес которой хранится в счетчике команд, выбирается очередная команда; при этом содержимое счетчика команд увеличивается на длину команды;
- 2) выбранная команда передается в устройство управления на регистр команд;
- 3) устройство управления расшифровывает адресное поле команды;
- 4) по сигналам УУ операнды считываются из памяти и записываются в АЛУ на специальные регистры операндов;
- 5) УУ расшифровывает код операции и выдает в АЛУ сигнал выполнить соответствующую операцию над данными;
- 6) результат операции либо остается в процессоре, либо отправляется в память, если в команде был указан адрес результата;
- 7) все предыдущие этапы повторяются до команды «стоп».

2.3.3. Архитектура компьютера

Архитектурой компьютера называется его описание на некотором общем уровне, включающее в себя описания пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т. д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного ЗУ, внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Наиболее распространены следующие архитектурные решения.

Классическая архитектура. Классическая архитектура (архитектура фон Неймана) — одно арифметико-логическое устройство, через которое проходит поток данных, и одно устройство управления, через которое проходит поток команд — программа (см. рис. 2.5). Это однопроцессорный компьютер. К этому типу архитектуры относится и архитектура персонального компьютера с общей шиной (рис. 2.7). Все функциональные блоки здесь связаны между собой общей шиной, называемой также системной магистралью, или системной шиной.

Системная магистраль (системная шина) включает в себя три многоуровневые шины: шину данных, шину адреса и шину управления. Шины представляют собой многопроводные линии.

Шина данных. По этой шине данные передаются между различными устройствами. Например, считанные из оперативной памяти данные могут быть переданы процессору для обработки, а затем полученные данные могут быть отправлены обратно в оперативную память для хранения. Таким образом, данные по шине

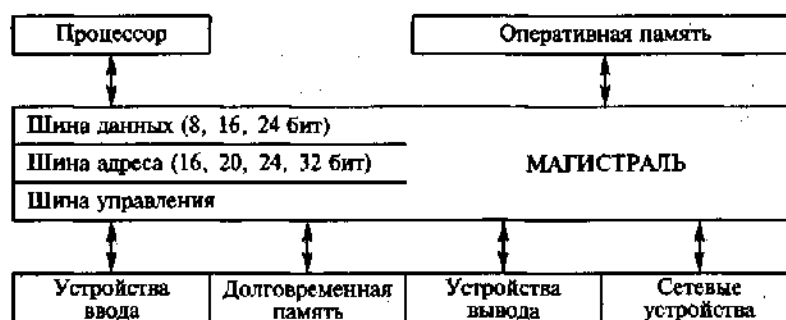


Рис. 2.7. Магистрально-модульное устройство компьютера

данных могут передаваться от устройства к устройству в любом направлении.

Разрядность шины данных определяется разрядностью процессора, т.е. числом двоичных разрядов, которые процессор обрабатывает за один такт. Разрядность процессоров постоянно увеличивается и в настоящее время составляет 64 бит.

Шина адреса. Выбор устройства, или ячейки памяти, куда пересылаются или откуда считываются данные по шине данных, производит процессор. Каждое устройство, или ячейка оперативной памяти, имеет свой адрес. Адрес передается по адресной шине, причем сигналы по ней передаются в одном направлении: от процессора к оперативной памяти и устройствам (однонаправленная шина).

Разрядность шины адреса определяет адресное пространство процессора, т.е. число ячеек оперативной памяти, которые могут иметь уникальные адреса. Число адресуемых ячеек памяти можно рассчитать по формуле

$$N = 2^i,$$

где i — разрядность шины адреса.

Разрядность шины адреса постоянно увеличивается и в современных персональных компьютерах составляет 32 бит. Таким образом, максимально возможное число адресуемых ячеек памяти

$$N = 2^{32} = 4\,294\,967\,396.$$

Шина управления. По шине управления передаются сигналы, определяющие характер обмена информацией по магистралам. Сигналы управления определяют, какую операцию, считывание или запись информации из памяти нужно производить; синхронизирует обмен информацией между устройствами и т.д.

Периферийные устройства (принтер и др.) подключаются к системной магистрали через специальные контроллеры.

Контроллер — устройство, которое связывает периферийное оборудование или каналы связи с системной магистралью центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

Существуют другие варианты построения однопроцессорных компьютеров, например применяемые известной компанией Apple Macintosh.

CISC-архитектура. Машины семейства IBM PC относятся к так называемой CISC-архитектуре компьютеров (CISC — Complete Instruction Set Computer — компьютер с полным набором команд). В системах команд процессоров, построенных по этой архитектуре, для каждого возможного действия предусмотрена отдельная команда.

Например, система команд процессора Intel Pentium состоит более чем из 1000 различных команд. Чем шире система команд, тем больше требуется бит памяти для кодирования каждой отдельной команды.

Если, например, система команд состоит всего из 4 действий, то для их кодирования требуется 2 бит памяти; для 8 возможных действий требуется 3 бит памяти; для 16 — 4 бит и т.д. Таким образом, расширение системы команд влечет за собой увеличение числа байт, выделяемых под одну машинную команду, а следовательно, и объема памяти, требуемой для записи всей программы в целом.

Кроме того, увеличивается среднее время выполнения одной машинной команды, а следовательно, и среднее время выполнения всей программы.

RISC-архитектура. В середине 1980-х гг. появились первые процессоры с сокращенной системой команд, построенные по так называемой RISC-архитектуре (RISC — Reduce Instruction Set Computer — компьютер с усеченной системой команд).

К архитектуре RISC относятся достаточно широко известные машины компании Apple Macintosh, которые имеют систему команд, обеспечивающую им в ряде случаев более высокую производительность по сравнению с машинами семейства IBM PC. Еще одно важное отличие этих машин состоит в том, что многие возможности, которые в семействе IBM PC обеспечиваются путем приобретения, установки и настройки дополнительного оборудования, в машинах семейства Macintosh являются встроенными и не требуют никакой настройки оборудования.

В качестве высокопроизводительных серверов достаточно часто используются машины семейств Sun Microsystems, Hewlett Packard и Compaq, которые также относятся к RISC-архитектуре. В качестве представителей других архитектур можно упомянуть еще и семейства переносных компьютеров классов Notebook (портативные).

Многопроцессорная архитектура. Наличие в компьютере нескольких процессоров означает, что параллельно может быть организовано много потоков данных и много потоков команд. Таким образом, параллельно могут выполняться несколько фрагментов одной задачи. Структурная схема такой машины, имеющей общую оперативную память и несколько процессоров, представлена на рис. 2.8.

Важной составной частью аппаратной реализации любого компьютера является процессор.

Центральный процессор в общем случае содержит:

- арифметико-логическое устройство;
- шины данных и шины адресов;
- регистры;
- счетчики команд;
- кэш — очень быструю память малого объема (от 8 до 512 Кбайт);
- математический сопроцессор чисел с плавающей точкой.

Современные процессоры выполняются в виде микропроцессоров.

Физически микропроцессор представляет собой интегральную схему — тонкую пластинку кристаллического кремния прямоугольной формы площадью всего несколько квадратных миллиметров, на которой размещены схемы, реализующие все функции процессора.

Кристалл-пластинка обычно помещается в пластмассовый или плоский керамический корпус и соединяется золотыми проводками с металлическими штырьками, чтобы его можно было присоединить к системной плате компьютера.

Память компьютера построена на триггерах (хранение одного бита информации реализуется с помощью триггера, см. подразд. 2.2), объединенных в группы по 8 бит, которые называются байтами (единицы измерения памяти совпадают с единицами измерения информации). Все байты пронумерованы. Номер байта называется его адресом.

Байты могут объединяться в ячейки, которые называются также словами. Для каждого компьютера характерна определенная длина слова — 2, 4 или 8 байт. Это не исключает использования ячеек памяти другой длины (например, полуслово, двойное слово).

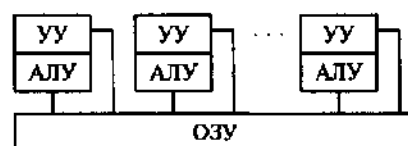


Рис. 2.8. Многопроцессорная архитектура

Таблица 2.16

Организация внутренней памяти

Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
ПОЛУСЛОВО		ПОЛУСЛОВО		ПОЛУСЛОВО		ПОЛУСЛОВО	
СЛОВО				СЛОВО			
ДВОЙНОЕ СЛОВО							

Как правило, в одном машинном слове может быть представлено либо одно целое число, либо одна команда. Однако допускаются переменные форматы представления информации. Разбиение памяти на слова для четырехбайтовых компьютеров представлено в табл. 2.16.

В основу архитектуры современных персональных компьютеров положен магистрально-модульный принцип построения, который позволяет потребителю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости ее модернизацию.

Такой подход еще называют принципом «открытой архитектуры». Если бы современный компьютер был сделан так же, как другие, существовавшие до его появления компьютеры, то он бы устарел через два-три года и мы бы давно уже о нем забыли. Действительно, кто сейчас помнит о моделях телевизоров, телефонов или даже автомобилей двенадцатилетней давности!

На основной электронной плате компьютера (системной, или материнской, плате) размещены только те блоки, которые осуществляют обработку информации (вычисления). Схемы, управляющие всеми остальными устройствами компьютера (монитором, дисками, принтером и т. д.), реализованы на отдельных платах, которые вставляются в стандартные разъемы на системной плате — слоты. К этим электронным схемам подводится электропитание из единого блока питания, а для удобства и надежности все это заключается в общий металлический или пластмассовый корпус — системный блок.

Наибольшую выгоду от открытости архитектуры получили, естественно, пользователи. Они получили возможность самостоятельно расширять возможности своих компьютеров, покупая соответствующие устройства и подсоединяя их в свободные разъемы на системной плате. При этом они не были связаны ассортиментом моделей, предлагаемых фирмой IBM, так как могли покупать дополнительные устройства, производимые независимыми фирмами. Они могли сэкономить деньги, ориентируясь при покупке компьютеров на свои сегодняшние, а не будущие потребности — ведь при необходимости компьютер можно модернизировать.

Контрольные вопросы

1. Перечислите основные устройства, обеспечивающие работу компьютера.
2. Что такое команда? Что описывает команда?
3. Сформулируйте функции памяти и процессора.
4. Что понимается под архитектурой компьютера?
5. В чем заключается принцип «открытой архитектуры»?
6. Почему именно работы американского математика Джона фон Неймана значительно изменили подходы к построению вычислительных устройств?
7. В чем состоит различие между адресным пространством процессора и объемом оперативной памяти, установленной в компьютере?
8. Найдите информацию об эволюции программ для компьютеров.

Глава 3

•

КОМПЬЮТЕР И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Возможности персонального компьютера определяются характеристиками его функциональных блоков. Современный рынок компьютерной техники очень разнообразен, поэтому довольно непросто выбрать нужное устройство или выбрать конфигурацию ПК с требуемыми характеристиками.

В этой главе приведены сведения об устройствах, входящих в состав компьютера и подключаемых к нему. Понимая, что технические характеристики устаревают очень быстро, основной упор сделан на принципы работы устройств.

В современном мире существует много задач, которые трудно или невозможно решить без организации информационной связи между различными компьютерами, поэтому в этой главе рассмотрены вопросы построения и функционирования компьютерных сетей, основные возможности, предоставляемые сетью Интернет.

3.1. АППАРАТНАЯ РЕАЛИЗАЦИЯ КОМПЬЮТЕРА

Обычно настольные компьютеры (рис. 3.1, а) состоят из трех частей (блоков): системного блока; клавиатуры; монитора.

Компьютеры выпускаются также в портативном и карманном исполнении (рис. 3.1, б, в).

В этих случаях системный блок, монитор и клавиатура заключены в один корпус (системный блок спрятан под клавиатурой, а монитор сделан как крышка к клавиатуре). Системный блок является главной составляющей компьютера, в нем располагаются все основные узлы:

- электронные схемы, управляющие работой компьютера (микропроцессор, оперативная память, контроллеры устройств и т. д.), расположенные на системной плате;
- блок питания, преобразующий электропитание сети в постоянный ток низкого напряжения, подаваемый на электронные схемы компьютера;
- накопители (или дисководы) для гибких магнитных и лазерных дисков, используемые для чтения и записи на гибкие магнитные диски (дискеты), и лазерные диски CD-R, CD-RW;

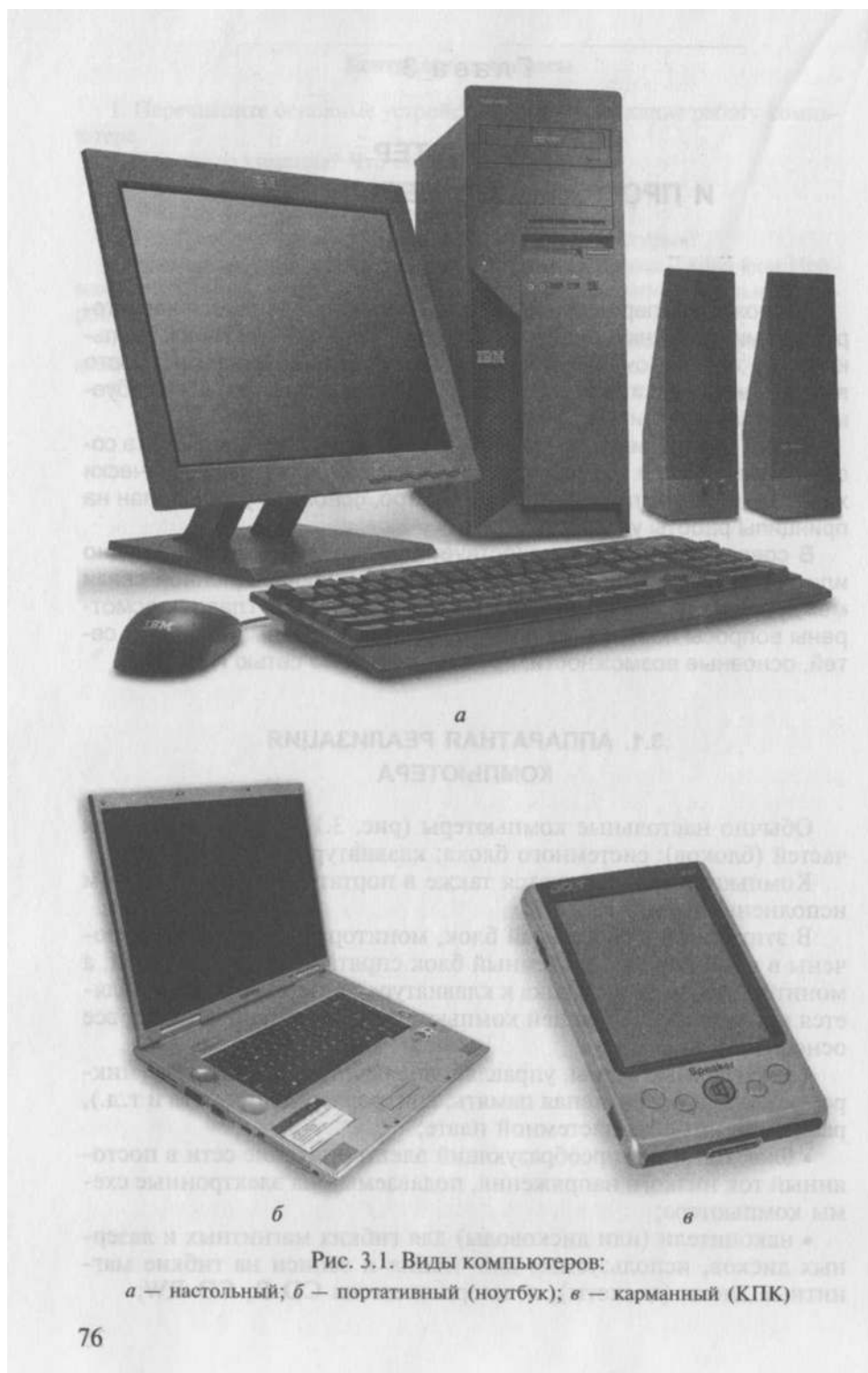




Рис. 3.2. Общая функциональная схема компьютера с подсоединенными периферийными устройствами

- накопитель на жестком магнитном диске, предназначенный для чтения и записи на несъемный жесткий магнитный диск (винчестер) и другие устройства.

На рис. 3.2 представлена общая функциональная схема компьютера.

3.1.1. Процессор (микропроцессор)

Центральный процессор (CPU — Central Processing Unit), или микропроцессор, — это основной функциональный компонент компьютера, который выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера.

Микропроцессор является «мозгом» компьютера. Он осуществляет дополнение программ, работающих на компьютере, и управляет работой остальных устройств компьютера. Скорость его работы во многом определяет быстродействие компьютера. В IBM PC используются микропроцессоры, разработанные фирмой Intel, а иногда — совместимые с ними микропроцессоры других фирм.

Важнейшей характеристикой процессора, определяющей его быстродействие, является его частота, т.е. число элементарных операций, производимых процессором за 1 с. За 25 лет тактовая частота процессора увеличилась в 375 раз: от 4 МГц (процессор 8086, 1978 г.) до 3 ГГц (процессор Pentium IV, 2003 г.) и более.

Другой характеристикой процессора, влияющей на его производительность, является разрядность. Разрядность процессора определяется числом двоичных разрядов, которые процессор обрабатывает за один такт. Разрядность процессора увеличилась за 20 лет в 8 раз. Часто уточняют разрядность процессора и пишут 64/32, это означает, что процессор имеет 64-разрядную шину данных и 32-разрядную шину адреса.

В первом отечественном школьном компьютере «Агат» (1985 г.) был установлен процессор, имевший разрядность 8/16. За один такт он обрабатывал 8 бит, а его адресное пространство составляло 64 Кбайт.

Современный процессор Pentium IV имеет разрядность 64/32, т.е. за один такт процессор обрабатывает 64 бит, а его адресное пространство составляет $2^{32} = 4\,294\,967\,296$ ячеек.

Производительность процессора зависит от особенностей архитектуры (наличие кэш-памяти и др.). Производительность процессора нельзя вычислить, она определяется в процессе тестирования по скорости выполнения процессором определенных операций в какой-либо программной среде.

3.1.2. Оперативная память

Оперативная память (ОЗУ) — память с произвольным доступом — это быстрое запоминающее устройство не очень большого объема, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Название «оперативная» память получила потому, что она работает очень быстро, так что процессору не приходится ждать при чтении данных из памяти или записи в память. Однако содержащиеся в ней данные сохраняются, пока компьютер включен; при выключении компьютера содержимое оперативной памяти стирается (за некоторыми исключениями).

Оперативная память (RAM — Random Access Memory — память с произвольным доступом) представляет собой множество ячеек, причем каждая ячейка имеет свой уникальный двоичный адрес (нумерация ячеек начинается с нуля). Каждая ячейка памяти имеет объем 1 байт, следовательно, максимальный объем адресуемой памяти для процессоров равен $4\,294\,967\,296$ байт = $4\,194\,304$ Кбайт = $= 4096$ Мбайт = 4 Гбайт.

В персональных компьютерах величина адресного пространства процессора и величина фактически установленной оперативной памяти практически всегда отличаются. Хотя объем адресуемой памяти может достигать 4 Гбайт, величина фактически установленной оперативной памяти может быть значительно меньше, например «всего» 512 Мбайт. Большинство современных компьютеров комплектуются модулями типа DIMM (Dual-In-line Memory Module — модуль памяти с двухрядным расположением микросхем). В компьютерных системах на самых современных процессорах используются высокоскоростные модули Rambus DRAM (RIMM), DDR DRAM, DDR DRAM II. Модули памяти характеризуются такими параметрами, как объем (128, 256, 512 Мбайт, 1 Гбайт), число микросхем, паспортная частота шины (от 100 до 533 МГц), время доступа к данным (от 74 нс) и число контактов (72, 168 или 184).

Кроме оперативной памяти существуют следующие типы внутренней памяти.

Кэш-память. Кэш-память (cache), или сверхоперативная память, — очень быстрое ЗУ небольшого объема, которое используется при обмене данными между процессором и оперативной памятью для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Кэш-памятью управляет специальное устройство — контроллер, который, анализируя выполняемую программу, пытается предвидеть, какие данные и команды, вероятнее всего, понадобятся в

ближайшее время процессору, и переписывает их в кэш-память. При этом возможны как попадания, так и промахи. В случае попадания, т. е. если в кэш записаны нужные данные, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше не найдена, то процессор считывает ее непосредственно из оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования. Кэш-память реализуется на микросхемах статической памяти SRAM (Static RAM), более быстродействующих, дорогих и малоемких, чем DRAM.

Современные микропроцессоры имеют встроенную кэш-память, так называемый кэш первого уровня, емкостью от 32 до 128 Кбайт. Кроме того, на системной плате компьютера может быть установлен кэш второго уровня емкостью 512 Кбайт и выше.

Постоянная память. Постоянная память (ROM — Read Only Memory — память только для чтения) — энергонезависимая память, используемая для хранения данных, которые никогда не потребуют изменения. Содержание памяти специальным образом «зашивается» в устройство при его изготовлении для постоянного хранения. Из постоянного запоминающего устройства (ПЗУ) данные можно только читать.

Прежде всего в постоянную память записывают программу управления работой самого процессора. В ПЗУ находятся программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Перепрограммируемая постоянная память. Перепрограммируемая постоянная память (Flash Memory) — энергонезависимая память, допускающая многократную перезапись своего содержимого.

Важнейшая микросхема постоянной памяти, или Flash-памяти, — модуль BIOS. BIOS (Basic Input/Output System — базовая система ввода-вывода) — совокупность программ, предназначенных для автоматического тестирования устройств после включения питания компьютера и загрузки операционной системы в оперативную память.

CMOS-память. CMOS RAM — это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Она используется для хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы. Содержимое CMOS изменяется специальной программой Setup, находящейся в BIOS.

Видеопамять. Видеопамять VRAM — разновидность оперативного ОЗУ, в котором хранятся закодированные изображения. Это ЗУ организовано так, что его содержимое доступно сразу двум устройствам — процессору и монитору, поэтому изображение на экране меняется одновременно с обновлением видеоданных в памяти.

3.1.3. Контроллеры и системная магистраль

Чтобы компьютер мог работать, необходимо, чтобы в его оперативной памяти находились программа и данные. А попадают они туда с различных устройств компьютера — клавиатуры, дисководов для магнитных дисков и других, которые называются внешними устройствами.

Таким образом, для работы компьютера необходим обмен информацией между оперативной памятью и внешними устройствами. Такой обмен называется вводом-выводом. Но этот обмен не происходит непосредственно, так как между любым внешним устройством и оперативной памятью в компьютере имеются два промежуточных звена.

1. Для каждого внешнего устройства в компьютере имеется электронная схема, которая им управляет. Эта схема называется контроллером, или адаптером. Некоторые контроллеры (например, контроллер дисков) могут управлять сразу несколькими устройствами.

2. Все контроллеры и адаптеры взаимодействуют с микропроцессором и оперативной памятью через системную магистраль передачи данных, которую обычно называют шиной (см. подразд. 2.3).

Быстродействие различных компонентов компьютера (процессора, оперативной памяти и контроллеров периферийных устройств) может существенно различаться.

В процессоре используется внутреннее умножение частоты, так как частота процессора больше, чем частота работы системной шины, по которой производится обмен информацией между процессором и оперативной памятью. Умножение частоты началось с процессора 486BX2-66, собственная частота которого составляет 66 МГц, а частота шины в два раза меньше — 33 МГц. В современных компьютерах частота процессора может превышать частоту системной шины в 10 раз (частота процессора 1 ГГц, а частота шины — 100 МГц). Частота работы контроллеров периферийных устройств (например, видео- и звуковой карт), наоборот, меньше частоты системной шины. Обмен информацией с такими устройствами производится по специальной шине PCI (Peripheral Component Interconnect bus — шина взаимодействия периферийных устройств). Если частота системной шины составляет 100 МГц, то частота шины PCI в 3 раза меньше — 33 МГц (в настоящее время частота системной шины составляет 400 МГц и более).

По мере увеличения разрешающей способности монитора и глубины цвета требования к быстродействию шины, связывающей видеоплату с процессором и оперативной памятью, возрастают. В настоящее время для подключения видеоплаты обычно используется специальная шина AGP (Accelerated Graphic Port — ускорен-

ный графический порт) с частотой в несколько раз большей, чем шина PCI.

Для того чтобы можно было соединить друг с другом различные устройства компьютера, они должны иметь одинаковый интерфейс (от *англ.* *inter* — между + *face* — лицо). Для согласования интерфейсов периферийные устройства подключаются к шине не напрямую, а через свои контроллеры (адаптеры) и порты.

Последовательные порты передают электрические импульсы, несущие информацию в машинном коде, последовательно один за другим. Обозначаются последовательные порты как COM1 и COM2, а аппаратно реализуются с помощью 25-контактного и 9-контактного разъемов, которые выведены на заднюю панель системного блока. К последовательным портам обычно подключаются мышь и модем.

Параллельный порт передает одновременно восемь электрических импульсов, несущих информацию в машинном коде. Обозначается параллельный порт как LPT, а аппаратно реализуется в виде 25-контактного разъема на задней стенке системного блока. Параллельный порт реализует более высокую скорость передачи информации, чем последовательные порты, и используется для подключения принтера.

Последовательный порт обменивается данными с процессором побайтно, а с внешними устройствами — побитно. Параллельный порт получает и посылает данные побайтно.

В последние годы широкое распространение получил порт USB (Universal Serial Bus — универсальная последовательная шина), который обеспечивает высокоскоростное подключение к компьютеру сразу нескольких периферийных устройств (сканеры, цифровые камеры и др.).

Для подключения к компьютеру дополнительных устройств может использоваться также Small Computer System Interface (интерфейс малых вычислительных систем). SCSI-адаптеры устанавливаются в слоты расширения системной платы и обеспечивают высокоскоростное подключение до семи различных устройств (винчестеров, сканеров, CD-ROM дисководов и др.). Для подключения джойстиков, предназначенных для управления играми, используется специальный Game-порт (игровой порт), который обычно размещается на звуковой плате.

3.1.4. Хранение информации. Внешняя память

Основной функцией внешней памяти компьютера является способность долговременно хранить большой объем информации (программы, документы, аудио- и видеоклипы и т.д.). Устройство, которое обеспечивает запись/считывание информации, на-

зывается накопителем, или дисководом, а хранится информация на носителях (например, на дискетах).

В качестве устройств внешней памяти используются накопители на гибких магнитных дисках (НГМД), жестких магнитных дисках (НЖМД), компакт-дисках, магнитной ленте (стримеры), Flash.

Накопители на гибких магнитных дисках. Гибкий диск (floppy disk), или дискета, — носитель небольшого объема информации, представляющий собой гибкий пластиковый диск в защитной оболочке. Он используется для переноса данных с одного компьютера на другой.

В процессе записи информации на гибкие и жесткие магнитные диски головка дисковода с сердечником из магнитомягкого материала (малая остаточная намагниченность) перемещается вдоль магнитного слоя магнито жесткого носителя (большая остаточная намагниченность); при этом на магнитную головку поступают последовательности электрических импульсов (последовательности логических единиц и нулей), которые создают в головке магнитное поле.

Магнитные домены на ферромагнитной поверхности материала выстраиваются вдоль дорожек в направлении приложенного магнитного поля своими северными и южными полюсами. Обычно устанавливается однозначное соответствие между двоичной информацией и ориентацией магнитных доменов. Такой способ записи двоичной информации на магнитной поверхности называется магнитным кодированием.

При считывании информации, наоборот, намагниченные участки носителя вызывают в магнитной головке импульсы тока (явление электромагнитной индукции). Последовательности таких импульсов передаются по магистрали в оперативную память компьютера.

Информация записывается по концентрическим дорожкам (трекам), которые делятся на секторы (рис. 3.3). Число дорожек и секторов зависит от типа и формата дискеты. Сектор хранит минимальную порцию информации, которая может быть записана на диск или считана с него. Емкость сектора постоянна и составляет 512 байт.

В настоящее время наибольшее распространение получили дискеты со следующими характеристиками:

- диаметр — 3,5 дюйма (89 мм);
- емкость — 1,44 Мбайт;
- число дорожек — 80;
- число секторов на дорожках — 18.

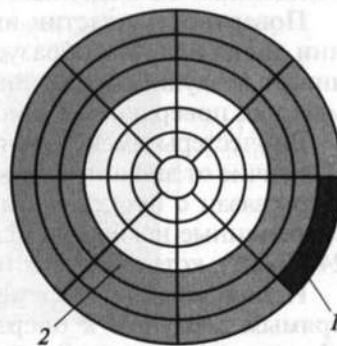


Рис. 3.3. Структура диска:

1 — сектор; 2 — трек

Дискета устанавливается в накопитель на гибких магнитных дисках (floppy-disk drive), автоматически в нем фиксируется, после чего механизм накопителя раскручивается до частоты вращения 360 мин^{-1} . В накопителе вращается сама дискета, магнитные головки остаются неподвижными.

Накопитель связан с процессором через контроллер гибких дисков. В последнее время появились трехдюймовые дискеты, которые могут хранить до 3 Гбайт информации. Они изготавливаются по новой технологии Nano2 и требуют специального оборудования для чтения и записи.

Для того чтобы на диске можно было хранить информацию, диск должен быть отформатирован, т.е. должна быть создана физическая и логическая структура диска (см. подразд. 4.2).

Накопители на жестких магнитных дисках. Если гибкие диски — это средство переноса данных между компьютерами, то жесткий диск — это информационная база компьютера.

Накопитель на жестких магнитных дисках (HDD — Hard Disk Drive), или винчестер, — это наиболее распространенное запоминающее устройство большой емкости, в котором носителями информации являются круглые алюминиевые пластины, обе поверхности которых покрыты слоем магнитного материала толщиной 1,1 мкм, а также слоем смазки для предохранения головки от повреждения при ее опускании и подъеме на ходу. Винчестер используется для постоянного хранения информации — программ и данных. Как и у дискеты, рабочие поверхности пластин разделены на кольцевые концентрические дорожки, а дорожки — на секторы.

Головки считывания-записи вместе с их несущей конструкцией и дисками заключены в герметически закрытый корпус, называемый модулем данных. При установке модуля данных на диск-вод он автоматически соединяется с системой, подкачивающей очищенный охлажденный воздух.

Поверхность пластин имеет магнитное покрытие. При вращении диска над ним образуется воздушный слой, который обеспечивает воздушную подушку для зависания головки на высоте 0,5 мкм над поверхностью диска.

Винчестеры имеют очень большую емкость — 50, 80, 100 Гбайт. В отличие от дискеты жесткий диск вращается непрерывно. Винчестер связан с процессором через контроллер жесткого диска. Все современные накопители снабжаются встроенным кэшем (обычно 24 Мбайт), который существенно повышает их производительность.

Использование современных контроллеров жестких дисков с прямым доступом к оперативной памяти UltraDMA-133 (Direct Memory Access — прямой доступ к памяти) позволяет достигать скорости обмена данными между жесткими дисками и оперативной памятью до 133 Мбайт/с (обычный показатель — 1012 байт/с).

Накопители на компакт-дисках. В настоящее время наибольшую популярность приобрели накопители на лазерных дисках (CD-ROM, DVD-ROM, CD-RW) (рис. 3.4, а).

В процессе считывания информации с лазерных дисков луч лазера, установленного в дисковом устройстве, падает на поверхность вращающегося диска и отражается. Так как поверхность лазерного диска имеет участки с различными коэффициентами отражения, то отраженный луч также меняет свою интенсивность (логические 0 или 1).

В процессе записи информации на лазерные диски для создания участков поверхности с различными коэффициентами отражения применяются различные технологии — от простой штамповки до изменения отражающей способности участков поверхности дисков мощным лазером.

Носителем информации является CD-ROM (Compact Disk Read-Only Memory — компакт-диск, с которого можно только читать). CD-ROM представляет собой прозрачный полимерный диск диаметром 12 см и толщиной 1,2 мм, на одну сторону которого на-



Рис. 3.4. CD-ROM и DVD-ROM:

а — внешний вид; б — запись информации на дисках CD-ROM и DVD-ROM

пылен светоотражающий слой алюминия, защищенный от повреждений слоем прозрачного лака. Толщина напыления составляет несколько десятитысячных долей миллиметра. Информация на диске представляется в виде последовательности впадин (углублений в диске) и выступов (их уровень соответствует поверхности диска), расположенных на спиральной дорожке, выходящей вблизи оси диска (рис. 3.4, б). На каждом дюйме* по радиусу диска размещается 16 тыс. витков спиральной дорожки. Для сравнения: на поверхности жесткого диска на каждом дюйме по радиусу помещается лишь несколько сотен дорожек. Емкость CD достигает 780 Мбайт. Информация наносится на диск при его изготовлении и не может быть изменена.

CD-ROM обладают высокой удельной информационной емкостью, что позволяет создавать на их основе справочные системы и учебные комплексы с большой иллюстративной базой.

CD-ROM просты и удобны в работе, имеют низкую стоимость, практически не изнашиваются, не могут быть поражены вирусами, с них невозможно случайно стереть информацию.

В отличие от магнитных дисков компакт-диски имеют не множество кольцевых дорожек, а одну — спиральную. В связи с этим угловая скорость вращения диска не постоянна. Она линейно уменьшается в процессе продвижения читающей лазерной головки к краю диска.

Для работы с CD-ROM нужно подключить к компьютеру накопитель CD-R, преобразующий последовательность углублений и выступов на поверхности CD-ROM в последовательность двоичных сигналов, в котором используется считывающая головка с микролазером и светодиодом. Глубина впадин на поверхности диска равна четверти длины волны лазерного света. Если в двух последовательных тактах считывания информации луч света лазерной головки переходит с выступа на дно впадины или обратно, то разность длин путей света в этих тактах меняется на полуволну, что вызывает усиление или ослабление совместно попадающих на светодиод прямого и отраженного от диска света. Если в последовательных тактах считывания длина пути света не меняется, то и состояние светодиода не меняется. В результате ток через светодиод образует последовательность двоичных электрических сигналов, соответствующих сочетанию впадин и выступов на дорожке.

На смену CD-ROM стремительно идет технология цифровых видеодисков DVD. Эти диски имеют тот же размер, что и обычные CD, но вмещают больше информации — от 4,7 Гбайт и более, так как информация может быть записана с двух сторон, в два слоя на каждой стороне, а сами дорожки имеют меньшую толщину.

* 1 дюйм = 2,54 см.

На таких дисках выпускаются мультимедийные игры и интерактивные видеофильмы отличного качества, позволяющие зрителю просматривать эпизоды под разными углами камеры, выбирать различные варианты окончания картины, знакомиться с биографиями снявшихся актеров, наслаждаться великолепным качеством звука.

Записывающий накопитель CD-ROM (Compact Disk Recordable) способен наряду с прочтением обычных компакт-дисков записывать информацию на специальные оптические диски емкостью 650 Мбайт и более (до 800 Мбайт). В дисках CD-R отражающий слой выполнен из золотой пленки. Между этим слоем и поликарбонатной основой расположен регистрирующий слой из органического материала, темнеющего при нагревании. В процессе записи лазерный луч нагревает выбранные точки слоя, которые темнеют и перестают пропускать свет к отражающему слою, образуя участки, аналогичные впадинам. Накопители CD-R благодаря сильному удешевлению приобретают все большее распространение.

Накопитель CD-RW (Compact Disk Rewriter/Writer) может использоваться для многократной записи информации на CD-ROM.

Накопители на магнитной ленте (стримеры). Стример (tape streamer) — устройство для резервного копирования больших объемов информации. В качестве носителя здесь применяются кассеты с магнитной лентой емкостью 8... 12 Гбайт и больше.

Встроенные в стример средства аппаратного сжатия позволяют автоматически уплотнять информацию перед ее записью и восстанавливать после считывания, что увеличивает объем сохраняемой информации. Недостатком стримеров является их сравнительно низкая скорость записи, поиска и считывания информации.

Flash. В настоящее время используются накопители Flash (рис. 3.5), имеющие встроенную карту памяти объемом от 64 Мб до 8 Гб, пользующиеся большим спросом благодаря своей компактности.

По оценкам специалистов, объем информации, фиксируемой на различных носителях, превышает 1 Эбайт в год (10^{18} байт/год). Примерно 80 % этой информации хранится в цифровой форме на магнитных и оптических носителях и только 20 % — на аналоговых носителях (бумага, магнитные ленты, фото- и киноплёнки).

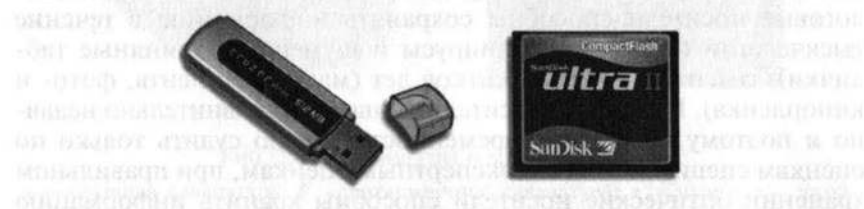


Рис. 3.5. Flash memory

Если всю записанную в 2000 г. информацию распределить на всех жителей планеты, то на каждого человека придется по 250 Мбайт, а для ее хранения потребуется 85 млн жестких магнитных дисков по 20 Гбайт.

Любой носитель информации характеризуется информационной емкостью, т.е. количеством информации, которое он может хранить. Наиболее информационно-емкими являются молекулы ДНК, которые имеют очень малый размер и большую плотность. Это позволяет хранить огромное количество информации, что дает возможность организму развиваться из одной единственной клетки, содержащей всю необходимую генетическую информацию.

Современные микросхемы памяти позволяют хранить в 1 мкм до 1010 бит информации, однако это в 100 млрд раз меньше, чем в ДНК. Можно сказать, что современные технологии пока существенно проигрывают биологической эволюции.

Однако если сравнивать информационную емкость традиционных носителей информации (книг) и современных компьютерных носителей, то прогресс очевиден. На каждом гибком магнитном диске может храниться книга объемом около 600 страниц, а на жестком магнитном диске — целая библиотека, включающая в себя десятки тысяч книг.

Большое значение имеет надежность и долговременность хранения информации. Надежность (устойчивость к повреждениям) достаточно высока у аналоговых носителей, повреждение которых приводит к потере информации только на поврежденном участке. Поврежденная часть фотографии не лишает возможности видеть оставшуюся часть, повреждение участка магнитной ленты приводит лишь к временному пропаданию звука и т.д.

Цифровые носители гораздо более чувствительны к повреждениям, даже утеря одного бита данных на магнитном или оптическом диске может привести к невозможности считать файл, т.е. к потере большого объема данных. Именно поэтому необходимо соблюдать правила эксплуатации и хранения цифровых носителей информации.

Наиболее долговременным носителем информации является молекула ДНК, которая в течение десятков тысяч (существования человека) и миллионов лет (существование некоторых живых организмов) сохраняет генетическую информацию данного вида. Аналоговые носители способны сохранять информацию в течение тысячелетий (египетские папирусы и шумерские глиняные таблички), сотен (бумага) и десятков лет (магнитная лента, фото- и киноплёнка). Цифровые носители появились сравнительно недавно и поэтому об их долговременности можно судить только по оценкам специалистов. По экспертным оценкам, при правильном хранении оптические носители способны хранить информацию сотни лет, а магнитные — десятки лет.

3.1.5. Устройства ввода информации

Клавиатура. Клавиатура компьютера (рис. 3.6, а, б) — устройство для ввода информации в компьютер и подачи управляющих сигналов. Оно содержит стандартный набор клавиш печатающей машинки и некоторые дополнительные клавиши: управляющие клавиши, функциональные клавиши, клавиши управления курсором, а также малую цифровую клавиатуру.

Курсор — светящийся символ на экране монитора, указывающий позицию, на которой будет отображаться следующий вводимый с клавиатуры знак. Все символы, набираемые на клавиатуре, немедленно отображаются на мониторе в позиции курсора.

Стандартная клавиатура имеет 101 клавишу и подключается к специальному разъему на системном блоке.



Рис. 3.6. Клавиатуры и манипуляторы:

а — обычная клавиатура; б — эргономичная клавиатура; в — мышь; г — джойстик; д — трекбол; е — тачпад

Клавиатура содержит встроенный микроконтроллер (местное устройство управления), который выполняет следующие функции:

- последовательно опрашивает клавиши, считывая введенный сигнал и вырабатывая двоичный скан-код клавиши;
- управляет световыми индикаторами клавиатуры;
- проводит внутреннюю диагностику неисправностей;
- осуществляет взаимодействие с центральным процессором через порт ввода-вывода клавиатуры.

Клавиатура имеет встроенный *буфер* — промежуточную память малого размера, куда помещаются введенные символы. В случае переполнения буфера нажатие клавиши будет сопровождаться звуковым сигналом — это означает, что символ не введен (отвергнут). Работу клавиатуры поддерживают специальные программы, «зашитые» в BIOS, а также драйвер клавиатуры, который обеспечивает возможность ввода русских букв, управление скоростью работы клавиатуры и др. (см. подразд. 3.3).

Все более популярными становятся клавиатуры на инфракрасных лучах, не требующие подключения к системному блоку. Передача сигналов с такой клавиатуры осуществляется по принципу, аналогичному дистанционному управлению.

Манипуляторы. Манипуляторы (мышь, джойстик, трекбол, тачпад, дигитайзер) — это специальные устройства, используемые для управления курсором.

Мышь имеет вид небольшой коробки, полностью помещающейся на ладони (рис. 3.6, а). Мышь связана с компьютером кабелем через специальный блок — адаптер, и ее движения преобразуются в соответствующие перемещения курсора по экрану дисплея. В оптико-механических манипуляторах основным функциональным элементом является массивный шар (металлический, покрытый резиной). У мыши он вращается при перемещении ее корпуса по горизонтальной поверхности, а у трекбола — вращается непосредственно рукой.

Вращение шара передается двум пластмассовым валам, положение которых с большой точностью считывается инфракрасными парами светодатчик—фотоприемник и затем преобразуется в электрический сигнал, управляющий движением указателя мыши на экране монитора.

Манипуляторы обычно имеют две кнопки управления, которые используются при работе с графическим интерфейсом программ, а также дополнительная кнопка-колесо, которая предназначена для прокрутки вверх или вниз не помещающегося целиком на экране изображения, текста или WEB-страницы.

Манипуляторы могут подключаться к компьютеру тремя различными способами: с использованием последовательного порта COM, специального маленького круглого 5-контактного разъема PS/2 и универсального USB-порта.

Джойстик — обычно это стержень-ручка, отклонение которой от вертикального положения приводит к передвижению курсора в соответствующем направлении по экрану монитора (рис. 3.6, з). Он часто применяется в компьютерных играх. В некоторых моделях в джойстик монтируется датчик давления: чем сильнее пользователь нажимает на ручку, тем быстрее движется курсор по экрану дисплея.

Трекбол — небольшая коробка с шариком, встроенным в верхнюю часть его корпуса (рис. 3.6, д). Пользователь рукой вращает шарик и перемещает соответственно курсор.

Тачпад (сенсорная панель) представляет собой панель прямоугольной формы, чувствительную к нажатию пальцев (рис. 3.6, е). Тачпад играет ту же роль, что и мышь, но является более компактным, не требующим пространственного перемещения устройства ввода. Он идеально подходит для портативных компьютеров. Тачпад встраивают непосредственно в клавиатуру для настольного компьютера. Прикоснувшись пальцем к поверхности тачпада и перемещая его, пользователь может маневрировать курсором так же, как и при использовании мыши. Нажатие на поверхность тачпада эквивалентно нажатию на кнопку мыши.

Дигитайзер — устройство для преобразования готовых изображений (чертежей, карт) в цифровую форму. Дигитайзер представляет собой плоскую панель — планшет, располагаемую на столе, и специальный инструмент — перо, с помощью которого указывается позиция на планшете. При перемещении пера по планшету фиксируются его координаты в близко расположенных точках, которые затем преобразуются в компьютере в требуемые единицы измерения.

Сканер. Сканер используется для оптического ввода в компьютер и преобразования в цифровую форму изображений (фотографий, рисунков, слайдов), а также текстовых документов.

Сканируемое изображение освещается белым светом (черно-белые сканеры) или тремя цветами (красным, зеленым и синим). Отраженный свет проецируется на линейку фотозлементов, которая движется, последовательно считывает изображение и преобразует его в компьютерный формат. Если при помощи сканера вводится текст, то компьютер воспринимает его как картинку, а не как последовательность символов. Для преобразования такого графического текста в обычный символьный формат используют программы оптического распознавания образов.

Такие программы способны распознавать текстовые документы на различных языках, представленные в различных формах (например, в виде таблиц) и с различным качеством печати (начиная с машинописных документов).

Существуют планшетные и ручные сканеры. Планшетные сканеры могут поставляться вместе со специальным слайд-модулем,



Рис. 3.7. Цифровая камера

предназначенным для сканирования слайдов. Разрешающая способность сканеров составляет от 300 dpi (от *англ.* dot per inch — точек на дюйм) и выше, т.е. на полоске изображения длиной 1 дюйм сканер может распознать 600, 800 и 1200 точек. Сканеры подключаются к компьютеру различными способами: с помощью SCSI-адаптеров, к параллельному или USB-портам компьютера.

Цифровые камеры и ТВ-тюнеры. В последние годы все большее распространение получают цифровые камеры (видеокамеры и фотоаппараты). Цифровые камеры позволяют получать видеоизображение и фотоснимки непосредственно в цифровом (компьютерном) формате.

Цифровые видеокамеры (рис. 3.7) могут быть постоянно подключены к компьютеру и обеспечивать запись видеоизображения на жесткий диск или его передачу по компьютерным сетям.

Цифровые фотоаппараты позволяют получать высококачественные фотографии, для хранения которых используются специальные модули памяти или жесткие диски очень маленького размера. Запись изображений на жесткий диск компьютера может осуществляться с помощью подключения камеры к USB-порту компьютера. Если установить в компьютер специальную плату (ТВ-тюнер) и подключить к ее входу телевизионную антенну, то появится возможность просматривать телевизионные передачи непосредственно на компьютере.

3.1.6. Устройства вывода информации

Монитор. Это устройство визуального отображения информации (в виде текста, таблиц, рисунков, чертежей и т.д.).

Монитор является универсальным устройством вывода информации и подключается к видеокарте, которая устанавливается в слот расширения системной платы в системном блоке.

Изображение в компьютерном формате (в виде последовательностей нулей и единиц) хранится в видеопамяти, размещенной на видеокарте. Изображение на экране монитора формируется путем считывания содержимого видеопамяти компьютера и отображения его на экран.

Мониторы на электронно-лучевой трубке (ЭЛТ) (рис. 3.8, а) используется в настольных компьютерах. Мониторы могут иметь различный размер экрана. Размер диагонали экрана измеряется в дюймах и обычно составляет 14, 15, 17 и более дюймов.

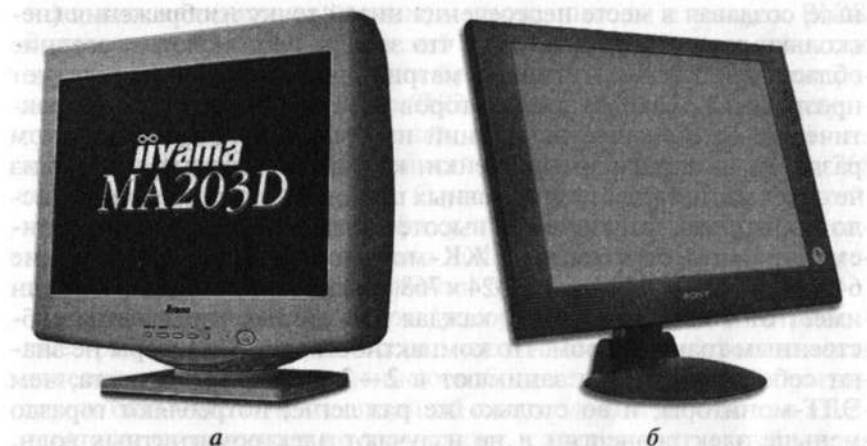


Рис. 3.8. Виды мониторов:

а — на электронно-лучевой трубке; *б* — жидкокристаллический

Размер экрана определяет разрешающую способность (число цветowych точек по вертикали и горизонтали) монитора в стандартном режиме: 14 дюймов — 640×480 ; 15 дюймов — 800×600 ; 17 дюймов — 1024×768 .

Основной элемент такого монитора — электронно-лучевая трубка. Изображение на экране монитора создается пучком электронов, излучаемых электронной пушкой.

В современных мониторах обновление изображения происходит обычно с частотой 75 и более раз в секунду, что обеспечивает комфортность восприятия изображения пользователем компьютера (человек не замечает мерцания изображения).

Жидкокристаллические (ЖК) мониторы (рис. 3.8, *б*) все шире используются наряду с традиционными ЭЛТ-мониторами. Жидкие кристаллы — это особое состояние некоторых органических веществ, в котором они обладают текучестью и свойством образовывать пространственные структуры, подобные кристаллическим. Жидкие кристаллы могут изменять свою структуру и светооптические свойства под действием электрического напряжения. Меняя с помощью электрического поля ориентацию групп кристаллов и используя введенные в жидкокристаллический раствор вещества, способные излучать свет под воздействием электрического поля, можно создавать высококачественные изображения, передающие более 15 млн цветowych оттенков.

Большинство ЖК-мониторов использует тонкую пленку из жидких кристаллов, помещенную между двумя стеклянными пластинами. Заряды передаются через так называемую пассивную матрицу — сетку невидимых нитей, горизонтальных и вертикаль-

ных, создавая в месте пересечения нитей точку изображения (несколько размытого из-за того, что заряды проникают в соседние области жидкости). Активные матрицы вместо нитей используют прозрачный экран из транзисторов и обеспечивают яркое, практически не имеющее искажений изображение. Экран при этом разделен на независимые ячейки, каждая из которых состоит из четырех частей (для трех основных цветов и одна резервная). Число таких ячеек по ширине и высоте экрана называют разрешением экрана. Современные ЖК-мониторы имеют разрешение 640 × 480, 1280 × 1024 или 1024 × 768 точек. Таким образом, экран имеет от 1 до 5 млн точек, каждая из которых управляется собственным транзистором. По компактности ЖК-мониторы не знают себе равных. Они занимают в 2—3 раза меньше места, чем ЭЛТ-мониторы, и во столько же раз легче; потребляют гораздо меньше электроэнергии и не излучают электромагнитных волн, отрицательно воздействующих на здоровье людей.

При работе некоторых программ или операционной системы MS-DOS экран монитора используется в текстовом режиме, т.е. экран монитора условно разбивается на отдельные участки — знакоместа, чаще всего на 25 строк по 80 символов (знакомест). В каждое знакоместо может быть выведен один из 256 заранее заданных символов. В число этих символов входят большие и малые латинские буквы, цифры, символы:

~ ! @ # \$ % ^ & * () _ + | ` ~ = \ { } [] ; : ' " < , > . ? / ,

а также псевдографические символы, используемые для вывода на экран таблиц и диаграмм, построения рамок вокруг участков экрана и т.д. (рис. 3.9).

В число символов, изображаемых на экране в текстовом режиме, могут входить и символы кириллицы (буквы русского алфавита). На цветных мониторах каждому знакоместу может соответствовать свой цвет символа и свой цвет фона.

Кроме монитора для получения качественного изображения важны и характеристики видеокарты.

Видеокарта — это электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Она

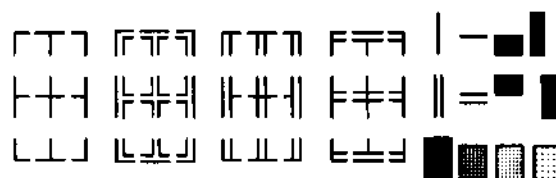


Рис. 3.9. Примеры построения рамок

содержит видеопамять, регистры ввода-вывода и модуль BIOS. Видеокарта посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения. Наиболее распространенная видеокарта — адаптер SVGA (Super Video Graphics Array — супервидеографический массив), который может отображать на экране дисплея 1280×1024 пикселей при 256 цветах и 1024×768 пикселей при 16 млн цветах. С увеличением числа приложений, использующих сложную графику и видео, наряду с традиционными видеоадаптерами широко используются разнообразные устройства компьютерной обработки видеосигналов:

- графические акселераторы (ускорители) — специализированные графические сопроцессоры, увеличивающие эффективность видеосистемы. Их применение освобождает центральный процессор от большого объема операций с видеоданными, так как акселераторы самостоятельно вычисляют, какие пиксели отображать на экране и каковы их цвета;

- фрейм-грабберы, которые позволяют отображать на экране компьютера видеосигнал от видеомagneтофона, камеры, лазерного проигрывателя, с тем чтобы захватить нужный кадр в память и впоследствии сохранить его в виде файла.

TV-тюнеры — видеоплаты, превращающие компьютер в телевизор.

Принтер. Это печатающее устройство, которое осуществляет вывод из компьютера закодированной информации в виде печатных копий текста или графики. Существуют тысячи наименований принтеров, но основных видов принтеров три: матричный, лазерный и струйный.

Матричный принтер (рис. 3.10, а) — это принтер ударного действия. Печатающая головка матричного принтера состоит из вертикального столбца маленьких игл (обычно 9 или 24), которые под воздействием магнитного поля выталкиваются из головки и ударяют по бумаге (через красящую ленту). Перемещаясь, печатающая головка оставляет на бумаге строку символов. Недостатками этих недорогих принтеров являются их шумная и медленная работа, а также невысокое качество печати.

Лазерный принтер (рис. 3.10, б) работает примерно так же, как копировальный аппарат. Компьютер формирует в своей памяти образ страницы текста и передает его принтеру. Информация о странице проецируется с помощью лазерного луча на вращающийся барабан со светочувствительным покрытием, меняющим электрические свойства в зависимости от освещенности.

После засветки на барабан, находящийся под электрическим напряжением, наносится красящий порошок — тонер, частицы которого налипают на засвеченные участки поверхности барабана. Принтер с помощью специального горячего валика протягивает бумагу под барабаном; тонер переносится на бумагу и «вплавляет-



Рис. 3.10. Принтеры:

а — матричный; *б* — лазерный; *в* — струйный; *г* — роликовый плоттер

ся» в нее, оставляя стойкое высококачественное изображение. Цветные лазерные принтеры пока еще очень дороги. Отличное типографское качество печати лазерных принтеров обеспечивается за счет высокой разрешающей способности, которая может достигать 1200 dpi и более. Это означает, что полоска изображения длиной в 1 дюйм формируется из 1200 точек.

Струйный принтер (рис. 3.10, *в*) генерирует символы в виде последовательности чернильных точек. Печатающая головка принтера имеет крошечные сопла, через которые на страницу выбрызгиваются быстросохнущие чернила. Эти принтеры требовательны к качеству бумаги. Цветные струйные принтеры создают цвета, комбинируя чернила четырех основных цветов: ярко-голубого, пурпурного, желтого и черного.

Принтер связан с компьютером при помощи кабеля, один конец которого вставляется своим разъемом в гнездо принтера, а

другой — в порт принтера компьютера. Порт — это разъем, через который можно соединить процессор компьютера с внешним устройством. Каждый принтер обязательно имеет свой драйвер — программу, которая способна переводить (транслировать) стандартные команды печати компьютера в специальные команды, требующиеся для каждого принтера.

Плоттер. Плоттер (графопостроитель, рис. 3.10, з) — устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера. Плоттеры используют для получения сложных конструкторских чертежей, архитектурных планов, географических и метеорологических карт, деловых схем. Плоттеры рисуют изображения с помощью пера.

Роликовые плоттеры прокручивают бумагу под пером, а планшетные плоттеры перемещают перо через всю поверхность горизонтально лежащей бумаги.

3.1.7. Устройства ввода-вывода информации

Звуковая карта. Это специальная электронная плата, которая позволяет записывать звук, воспроизводить и создавать его программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования (рис. 3.11).

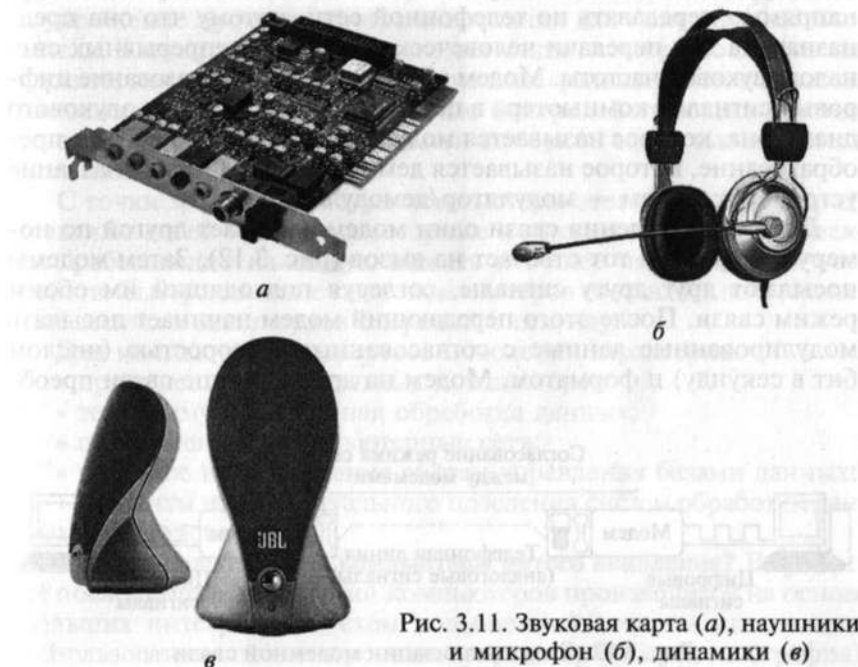


Рис. 3.11. Звуковая карта (а), наушники и микрофон (б), динамики (в)

Звуковая карта содержит два преобразователя информации:

- аналого-цифровой, который преобразует непрерывные (т.е. аналоговые) звуковые сигналы (речь, музыку, шум) в цифровой двоичный код и записывает его на магнитный носитель;
- цифроаналоговый, выполняющий обратное преобразование сохраненного в цифровом виде звука в аналоговый сигнал, который затем воспроизводится с помощью акустической системы, синтезатора звука или наушников.

Профессиональные звуковые платы позволяют выполнять сложную обработку звука, обеспечивают стереозвучание, имеют собственное ПЗУ с хранящимися в нем сотнями тембров звучаний различных музыкальных инструментов. Звуковые файлы обычно имеют очень большие размеры. Так, трехминутный звуковой файл со стереозвучанием занимает примерно 30 Мбайт памяти. Поэтому платы Sound Blaster помимо своих основных функций обеспечивают автоматическое сжатие файлов.

Область применения звуковых плат — компьютерные игры, обучающие программные системы, рекламные презентации, «голосовая почта» (voice mail) между компьютерами, озвучивание различных процессов, происходящих в компьютерном оборудовании, таких как отсутствие бумаги в принтере и т.п.

Модем. Это устройство для передачи компьютерных данных на большие расстояния по телефонным линиям связи.

Цифровые сигналы, вырабатываемые компьютером, нельзя напрямую передавать по телефонной сети, потому что она предназначена для передачи человеческой речи — непрерывных сигналов звуковой частоты. Модем обеспечивает преобразование цифровых сигналов компьютера в переменный ток частоты звукового диапазона, которое называется модуляцией, а также обратное преобразование, которое называется демодуляцией. Отсюда название устройства: модем — модулятор/демодулятор.

Для осуществления связи один модем вызывает другой по номеру телефона, а тот отвечает на вызов (рис. 3.12). Затем модемы посылают друг другу сигналы, согласуя подходящий им обоим режим связи. После этого передающий модем начинает посылать модулированные данные с согласованной скоростью (числом бит в секунду) и форматом. Модем на другом конце связи преоб-

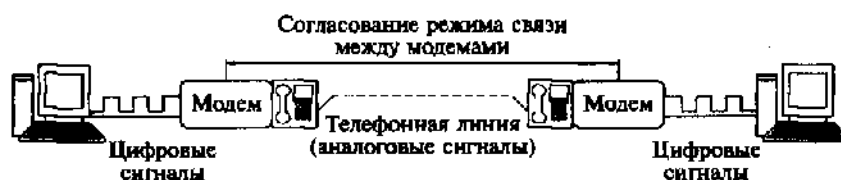


Рис. 3.12. Схема реализации модемной связи

разует полученную информацию в цифровой вид и передает ее своему компьютеру (рис. 3.12). Закончив сеанс связи, модем отключается от линии. Управление модемом осуществляется с помощью специального коммутационного программного обеспечения.

Модемы бывают внешние, выполненные в виде отдельного устройства, и внутренние, представляющие собой электронную плату, установленную внутри компьютера. Все модемы поддерживают и функции факсов.

Факс. Это устройство факсимильной передачи изображения по телефонной сети. Название «факс» произошло от слова «факсимиле» (от лат. *fac simile* — сделай подобное), означающего точное воспроизведение графического оригинала (подписи, документа и т.д.) средствами печати.

3.1.8. Компьютеры четвертого и пятого поколения

Компьютеры четвертого поколения. Это поколение компьютерной техники, разработанное после 1980 г. Наиболее важный в концептуальном отношении критерий, по которому эти компьютеры можно отделить от машин третьего поколения, состоит в том, что машины четвертого поколения проектировались в расчете на эффективное использование современных высокоуровневых языков и упрощение процесса программирования для конечного пользователя. В аппаратном отношении для машин четвертого поколения характерно широкое использование интегральных схем в качестве элементной базы, а также наличие быстродействующих запоминающих устройств с произвольной выборкой емкостью в десятки мегабайт.

С точки зрения структуры компьютеры этого поколения представляют собой многопроцессорные и многомашинные комплексы, работающие на общую память и общее поле внешних устройств. Быстродействие таких компьютеров составляет до нескольких десятков миллиардов операций в секунду.

Для компьютеров четвертого поколения характерны:

- применение персональных компьютеров;
- телекоммуникационная обработка данных;
- объединение в компьютерные сети;
- широкое использование систем управления базами данных;
- элементы интеллектуального поведения систем обработки данных и устройств.

Какими должны быть компьютеры пятого поколения? Разработка последующих поколений компьютеров производится на основе больших интегральных схем повышенной степени интеграции, использования оптоэлектронных принципов (лазеры, голография).

Развитие идет также по пути «интеллектуализации» компьютеров, устранения барьера между человеком и компьютером. Компьютеры будут способны воспринимать информацию с рукописного или печатного текста, с бланков; узнавать пользователя по голосу; осуществлять перевод с одного языка на другой.

В компьютерах пятого поколения произойдет качественный переход от обработки данных к обработке знаний.

Архитектура компьютеров будущего поколения будет содержать два основных блока. Один из них — это традиционный компьютер, но теперь он лишен связи с пользователем. Эту связь осуществляет блок, так называемый интеллектуальный интерфейс. Его задача — понять текст, написанный на естественном языке и содержащий условие задачи, и перевести его в работающую программу для компьютера.

Будет также решаться проблема децентрализации вычислений с помощью компьютерных сетей, как больших, находящихся на значительном расстоянии друг от друга, так и миниатюрных компьютеров, размещенных на одном кристалле полупроводника.

3.1.9. Безопасная работа на компьютере

За последние годы компьютер превратился из устройства для профессионалов в самый обыденный предмет. За всю историю человек еще не имел дела с таким воздействием, режимом нагрузок, как при работе с компьютером. Организм человека и его сознание просто не успели адаптироваться. Как правильно выбрать компьютер и программы, как пользоваться компьютером, чтобы не навредить здоровью?

Важно выбрать оптимальную конфигурацию компьютера, качественное программное обеспечение и правильно организовать рабочее место. Кроме того, есть специфические правила работы на компьютере, соблюдение которых поможет вам избежать многих проблем.

Конфигурация компьютера должна соответствовать решаемым задачам.

Подобрать качественное программное обеспечение, правильно его настроить еще важнее, чем купить хорошее «железо», поэтому следует обратить внимание не только на функциональные возможности и стоимость программ, но и на то, насколько они удобны в работе, не утомляют ли глаза, каковы возможности их настройки.

Монитор больше всего влияет на здоровье. Современные мониторы стали безопаснее для здоровья, но еще не полностью.

Приведем параметры монитора, которые влияют на здоровье человека.

1. Частота строчной развертки (частота кадров). Изображение на мониторе всегда немного мерцает. Это мерцание приводит к утомлению глаз, головным болям, другим проблемам. Чем выше частота кадров, тем меньше мерцает экран. Стандарты на частоту кадров постоянно меняются. Частота развертки для того разрешения, с которым вы работаете, должна быть не ниже 85 Гц (для ЖК-монитора — 70 Гц). Эта частота должна поддерживаться и монитором, и графической платой.

2. Размер точки (точнее — шаг точки). При маленьком шаге точки изображение получается более четкое, глаза меньше утомляются. У разных мониторов может быть указан либо шаг точки (dot pitch), либо шаг полосы (stripe pitch). Шаг полосы в 0,25 мм приблизительно соответствует шагу точки в 0,28 мм.

3. Излучение монитора. Существуют стандарты безопасности мониторов. Принятый в 1990 г. стандарт MPR-II установил конкретные ограничения на электрическое и магнитное излучения. Еще более жесткие требования были установлены стандартами TCO-92 и TCO-95, в настоящее время мониторы должны удовлетворять стандарту TCO-99.

На экране монитора не должно быть бликов и отражений.

Чтобы монитор использовал свои возможности, в компьютере должна быть хорошая видеокарта. Напомним, что она обеспечивает необходимую частоту развертки и четкость изображения. Еще одна причина низкой эргономичности монитора — неправильная настройка компьютеров.

Потеря информации, безусловно, занимает первое место среди компьютерных стрессоров (если не по частоте, то по силе стресса).

Необходимость сохранять информацию — это второе (после дублирования информации) важнейшее правило работы на компьютере. Практически все компьютерные программы вводимую информацию держат в оперативной памяти, а на диск записывают только тогда, когда вы закрываете программу или явно указываете программе сохранить данные. При зависании (выключении, перезагрузке) компьютера данные из оперативной памяти пропадают, поэтому необходимо сохранять вводимые вами данные, завершив очередной этап работы. Можно включить режим автосохранения файлов, который есть в большинстве текстовых редакторов и в ряде других программ. Данные будут автоматически записываться на диск каждые несколько минут (интервал задается при настройке режима автосохранения). При сбоях электропитания необходимо приобрести устройство бесперебойного питания — UPS, которое позволяет при отключении электричества поработать еще несколько минут, чего вполне достаточно для того, чтобы корректно завершить работу всех программ, сохранить данные и выключить компьютер.

Любой жесткий диск, даже произведенный самой лучшей фирмой, рано или поздно отказывает, поэтому, во-первых, нельзя пренебрегать важнейшим правилом работы на компьютере — регулярно делать резервные копии данных, а во-вторых, необходимо регулярно тестировать диск (см. подразд. 4.2.4, 4.2.5).

Как правильно работать на компьютере. Очень важная задача — правильно организовать рабочее место. Удачное расположение его компонентов, правильная поза при работе за компьютером позволят серьезно уменьшить неприятные последствия для здоровья.

Основные факторы вредного воздействия на здоровье человека при работе с ПК:

- сидячее положение в течение длительного времени;
- воздействие электромагнитного излучения монитора;
- утомление глаз, нагрузка на зрение;
- перегрузка суставов кистей;
- стресс при потере информации.

Контрольные вопросы

1. Какие основные классы компьютеров вам известны?
2. Как конструктивно выполнены современные процессоры?
3. Перечислите виды внутренней памяти.
4. Сравните устройства внешней памяти, выделите их достоинства и недостатки.
5. В чем заключается вредное воздействие монитора на человека?
6. Какие существуют типы принтеров и какие принципы печати в них используются?
7. Чем плоттер отличается от принтера?
8. Какие функции будут расширены в компьютерах пятого поколения?
9. Сформулируйте причины успеха персональных компьютеров.
10. Какие устройства используются для перевода символов чисел, графического изображения и звука в наборы двоичных символов? (Используйте материал подразд. 1.4 и 2.1.)
11. В чем выражается ограниченность области применения персональных компьютеров?
12. Как бы вы комплектовали компьютер для офиса, дома, магазина, туристической фирмы?
13. Перечислите факторы вредного воздействия на здоровье человека при работе с ПК.
14. Какие меры безопасности необходимо применить при работе с ПК?

3.2. КОМПЬЮТЕРНЫЕ СЕТИ

Современное производство требует высоких скоростей обработки информации, удобных форм ее хранения и передачи. Необходимо также иметь динамичные способы обращения к ин-

формации, способы поиска данных в заданные временные интервалы, реализовывать сложную математическую и логическую обработку данных. Управление крупными предприятиями, управление экономикой на уровне страны требуют участия в этом процессе достаточно крупных коллективов. Такие коллективы могут располагаться в различных районах города, регионах страны и даже в различных странах. Для решения задач управления, обеспечивающих реализацию экономической стратегии, становятся важными и актуальными скорость и удобство обмена информацией, а также возможность тесного взаимодействия всех лиц, участвующих в процессе выработки управленческих решений.

Для реализации поставленных задач применяется распределенная обработка данных — обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему. Для реализации распределенной обработки данных были созданы многомашинные ассоциации, структура которых разрабатывается по одному из следующих направлений:

- многомашинные вычислительные комплексы (МВК);
- компьютерные (вычислительные) сети (КС).

Многомашинный вычислительный комплекс — группа установленных рядом вычислительных машин, объединенных с помощью специальных средств сопряжения и выполняющих совместно решение единой задачи.

Компьютерная (вычислительная) сеть — совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

Задачи, которые трудно или невозможно решить без организации информационной связи между различными компьютерами:

- перенос информации на большие расстояния (сотни, тысячи километров);
- совместное использование несколькими компьютерами дорогостоящих аппаратных, программных или информационных ресурсов: мощного процессора, емкого накопителя, высокопроизводительного лазерного принтера, баз данных, программного обеспечения и т. д.;
- совместная работа над большим проектом, когда исполнители должны всегда иметь последние (актуальные) копии общих данных во избежание путаницы и т. д.

Абоненты сети — объекты, генерирующие или потребляющие информацию в сети.

Станция — аппаратура, которая выполняет функции, связанные с передачей и приемом информации.

Абонентская система — совокупность абонента и станции.

Физическая передающая среда — линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных.

На базе физической передающей среды строится коммуникационная сеть, которая обеспечивает передачу информации между абонентскими системами.

При организации связи между двумя компьютерами за одним компьютером закрепляется роль поставщика ресурсов (программ, данных и т. д.), а за другим — роль пользователя этих ресурсов. В этом случае первый компьютер называется *сервером*, а второй — *клиентом*, или рабочей станцией. Работать можно только на компьютере-клиенте под управлением специального программного обеспечения.

Сервер (от англ. *serve* — обслуживать) — это высокопроизводительный компьютер с большим объемом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования).

Клиент (рабочая станция) — любой компьютер, имеющий доступ к услугам сервера.

Компьютерные сети — это совокупность технических, коммуникационных и программных средств, обеспечивающих эффективное распределение вычислительных ресурсов.

3.2.1. Классификация компьютерных сетей

В зависимости от территориального расположения абонентских систем вычислительные сети можно подразделить на три основных класса:

- глобальные (WAN — Wide Area Network);
- региональные (MAN — Metropolitan Area Network);
- локальные (LAN — Local Area Network).

Глобальная вычислительная сеть объединяет абонентов, расположенных в различных странах, на различных континентах. Взаимодействие между абонентами такой сети может осуществляться на базе телефонных линий связи, радиосвязи и систем спутниковой связи. Глобальные вычислительные сети позволят решить проблему объединения информационных ресурсов всего человечества и организации доступа к этим ресурсам.

Региональная вычислительная сеть связывает абонентов, расположенных на значительном расстоянии друг от друга. Она может включать в себя абонентов внутри большого города, экономического региона, отдельной страны. Обычно расстояние между абонентами региональной вычислительной сети составляет десятки — сотни километров.

Локальная вычислительная сеть (ЛВС) объединяет абонентов, расположенных в пределах небольшой территории. В настоящее время не существует четких ограничений на территориальный разброс абонентов локальной вычислительной сети. Обычно такая сеть привязана к конкретному месту. К классу локальных вычислительных сетей относятся сети отдельных предприятий, фирм, банков, офисов и т.д. Протяженность такой сети можно ограничить 2...2,5 км.

Объединение глобальных, региональных и локальных вычислительных сетей позволяет создавать многосетевые иерархии. Они обеспечивают мощные, экономически целесообразные средства обработки огромных информационных массивов и доступ к неограниченным информационным ресурсам. На рис. 3.13 приведена одна из возможных иерархий компьютерных сетей. Локальные вычислительные сети могут входить как компоненты в состав региональной сети, региональные сети — объединяться в составе глобальной сети и, наконец, глобальные сети могут также образовывать сложные структуры.

Компьютерная сеть Интернет является наиболее популярной глобальной сетью. В ее состав входит множество свободно соединенных сетей. Внутри каждой сети, входящей в Интернет, существует конкретная структура связи и определенная дисциплина управления. Внутри сети Интернет структура и методы соединений между различными сетями для конкретного пользователя не имеют никакого значения.

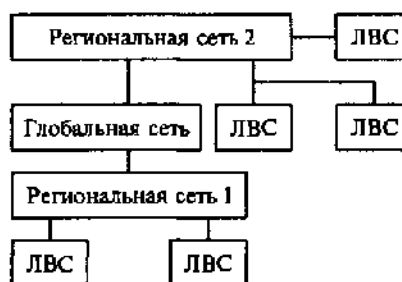


Рис. 3.13. Иерархия компьютерных сетей

3.2.2. Топология сети

Компьютерную сеть представляют как совокупность узлов (компьютеров и сетевого оборудования) и соединяющих их ветвей (каналов связи).

Различают узлы оконечные, расположенные в конце только одной ветви, промежуточные, расположенные на концах более чем одной ветви; смежные, соединенные, по крайней мере, одним путем, не содержащим никаких других узлов.

Компьютеры могут объединяться в сеть разными способами. Логический и физический способ соединения компьютеров, кабелей и других компонентов, в целом составляющих сеть, называется ее *топологией*. Топология характеризует свойства сетей, не

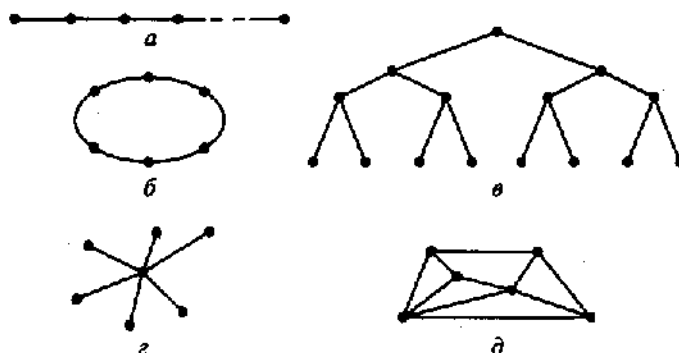


Рис. 3.14. Виды топологий сетей:

a — линейная; *б* — кольцевая; *в* — иерархическая; *г* — звездообразная; *д* — ячеистая

зависящие от их размеров. При этом не учитываются производительность и принцип работы этих объектов, их типы, длины каналов, хотя при проектировании эти факторы очень важны.

Наиболее распространены следующие виды топологий сетей (рис. 3.14):

- линейная, содержащая только два конечных узла, любое число промежуточных узлов и имеющая только один путь между любыми двумя узлами;
- кольцевая, в которой к каждому узлу присоединены только две ветви;
- древовидная, содержащая более двух конечных узлов и, по крайней мере, два промежуточных узла и имеющая между двумя узлами только один путь;
- звездообразная, имеющая только один промежуточный узел;
- ячеистая, содержащая, по крайней мере, два узла и имеющая два или более пути между ними. Это полносвязанная сеть, в которой имеется ветвь между любыми двумя узлами.

3.2.3. Способы коммутации и передачи данных

Основная функция систем передачи данных заключается в организации быстрой и надежной передачи информации произвольным абонентам сети, а также в сокращении затрат на передачу данных. Последнее особенно важно, так как за прошедшее десятилетие произошло увеличение доли затрат на передачу данных в общей структуре затрат на организацию сетевой обработки информации. Это объясняется главным образом тем, что затраты на техническое обеспечение сетей сократились за этот период примерно в 10 раз, тогда как затраты на организацию и эксплуата-

цию каналов связи — только в два раза. Важнейшая характеристика сетей передачи данных — время доставки информации — зависит от структуры сети передачи данных, пропускной способности линий связи, а также от способа соединения каналов связи между взаимодействующими абонентами сети и способа передачи данных по этим каналам. В настоящее время различают системы передачи данных с постоянным включением каналов связи (некоммутируемые каналы связи) и коммутацией на время передачи информации по этим каналам. При использовании некоммутируемых каналов связи средства приема-передачи абонентских пунктов и ПК постоянно соединены между собой, т.е. находятся в режиме on-line. В этом случае отсутствуют потери времени на коммутацию, обеспечиваются высокая степень готовности системы к передаче информации, более высокая надежность каналов связи и, как следствие, достоверность передачи информации. Недостатками такого способа организации связи являются низкий коэффициент использования аппаратуры передачи данных и линий связи, высокие расходы на эксплуатацию сети. Рентабельность подобных сетей достигается только при условии достаточно полной загрузки этих каналов.

Коммутация каналов. Способ коммутации каналов заключается в установлении физического канала связи для передачи данных непосредственно между абонентами сети.

Установление соединения состоит в том, что абонент посылает в канал связи заданный набор символов, прохождение которых по сети через соответствующие узлы коммутации вызывает установку нужного соединения с вызываемым абонентом. Этот транзитный канал образуется в начале сеанса связи, остается фиксированным на период передачи всей информации и разрывается только после завершения передачи информации. Такой способ соединения используется в основном в сетях, в которых требуется обеспечить непрерывность передачи сообщений.

Коммутация сообщений. При коммутации сообщений поступающая на узел связи информация передается в память узла связи, после чего анализируется адрес получателя. В зависимости от занятости требуемого канала сообщение либо передается в память соседнего узла, либо становится в очередь для последующей передачи. Таким образом, способ коммутации сообщений обеспечивает поэтапный характер передачи информации. Метод коммуникации сообщений обеспечивает независимость работы отдельных участков сети, что значительно повышает эффективность использования каналов связи при передаче одного и того же объема информации (которая, в этом случае может достигать 80...90 % от максимального значения). В системе с коммутацией сообщений происходит сглаживание несогласованности в пропускной способности каналов и более эффективно реализуется пере-

дача многоадресных сообщений (так как не требуется одновременного освобождения всех каналов между узлом-передатчиком и узлом-приемником). Передача информации может производиться в любое время, так как прямая связь абонентов друг с другом необязательна.

Коммутация пакетов. Этот способ сочетает в себе ряд преимуществ методов коммутации каналов и коммутации сообщений. При коммутации пакетов перед началом передачи сообщение разбивается на короткие пакеты фиксированной длины, которые затем передаются по сети. В пункте назначения эти пакеты вновь объединяются в первоначальное сообщение, а так как их длительное хранение в запоминающем устройстве узла связи не предполагается, пакеты передаются от узла к узлу с минимальной задержкой во времени. В этом отношении указанный метод близок методу коммутации каналов.

При коммутации пакетов их фиксированная длина обеспечивает эффективность обработки пакетов, предотвращает блокировку линий связи и значительно уменьшает емкость требуемой промежуточной памяти узлов связи. Кроме того, сокращается время задержки при передаче информации, т. е. скорость передачи информации выше, чем при методе коммутации сообщений.

3.2.4. Характеристики коммуникационной сети

Для оценки качества коммуникационной сети можно использовать следующие характеристики:

- скорость передачи по каналу связи;
- пропускная способность канала связи;
- достоверность передачи информации;
- надежность канала связи и модемов.

Скорость передачи данных по каналу связи измеряется числом бит информации, передаваемых за единицу времени — секунду, поэтому единица измерения скорости передачи данных — бит в секунду (бит/с).

Пропускная способность оценивается количеством знаков, передаваемых по каналу за единицу времени — секунду. При этом в состав сообщения включаются и все служебные символы. Теоретическая пропускная способность определяется скоростью передачи данных. Реальная пропускная способность зависит от ряда факторов, среди которых и способ передачи, и качество канала связи, и условия его эксплуатации, и структура сообщений. Единица измерения пропускной способности канала связи — знак в секунду (знак/с).

Существенной характеристикой коммуникационной системы любой сети является достоверность передаваемой информации.

Требуемый уровень достоверности должны обеспечивать как аппаратура, так и канал связи. Нецелесообразно использовать дорогостоящую аппаратуру, если относительно уровня достоверности канал связи не обеспечивает необходимых требований. Единица измерения достоверности — число ошибок на знак (ошибка/знак).

Для вычислительных сетей этот показатель должен находиться в пределах $10^{-6} \dots 10^{-7}$ ошибок/знак, т.е. допускается одна ошибка на миллион или десять миллионов переданных знаков.

3.2.5. Архитектура компьютерной сети

Архитектура сети — это реализованная структура сети передачи данных, определяющая ее топологию, состав устройств и правила их взаимодействия в сети. В рамках архитектуры сети рассматриваются вопросы кодирования информации, ее адресации и передачи, управления потоком сообщений, контроля ошибок и анализа работы сети в аварийных ситуациях и при ухудшении характеристик.

Наиболее распространены следующие архитектуры сети:

- Ethernet (от *англ.* ether — эфир) — широковещательная сеть. Это значит, что все станции сети могут принимать все сообщения. Топология — линейная или звездообразная. Скорость передачи данных — 10 или 100 Мбит/с. Arcnet (Attached Resource Computer Network — компьютерная сеть соединенных ресурсов) — широковещательная сеть. Физическая топология — дерево. Скорость передачи данных — 2,5 Мбит/с;

- Token Ring (эстафетная кольцевая сеть, сеть с передачей маркера) — кольцевая сеть, в которой принцип передачи данных основан на том, что каждый узел кольца ожидает прибытия некоторой короткой уникальной последовательности битов — маркера — из смежного предыдущего узла. Поступление маркера указывает на то, что можно передавать сообщение из данного узла дальше по ходу потока. Скорость передачи данных — 4 или 16 Мбит/с;

- FDDI (Fiber Distributed Data Interface) — сетевая архитектура высокоскоростной передачи данных по оптоволоконным линиям. Скорость передачи данных — 100 Мбит/с. Топология — двойное кольцо или смешанная (с включением звездообразных или древовидных подсетей). Максимальное число станций в сети — 1000. Очень высокая стоимость оборудования;

- ATM (Asynchronous Transfer Mode) — перспективная, пока еще очень дорогая архитектура, обеспечивающая передачу цифровых данных, видеoinформации и голоса по одним и тем же линиям. Скорость передачи данных — до 2,5 Гбит/с. Линии связи оптические.

3.2.6. Аппаратные средства организаций компьютерной сети

Для соединения устройств в сети используется специальное оборудование.

Сетевые кабели подразделяются на коаксиальные, состоящие из двух изолированных между собой концентрических проводников, из которых внешний имеет вид трубки; оптоволоконные, состоящие из стеклянной трубки, внутри которой находятся оптические волокна, представляющие практически идеальную передающую среду; кабели на витых парах, образованные двумя переплетенными друг с другом проводами, и др.

Коннекторы (соединители) служат для подключения кабелей к компьютеру, а разъемы — для соединения отрезков кабеля.

Сетевые интерфейсные адаптеры служат для приема и передачи данных. Эти же функции выполняет сетевая карта. В соответствии с определенным протоколом они управляют доступом к среде передачи данных. Размещаются сетевые интерфейсные адаптеры в системных блоках компьютеров, подключенных к сети. К разъемам адаптеров подключается сетевой кабель.

Трансиверы повышают уровень качества передачи данных по кабелю, отвечают за прием сигналов из сети и обнаружение конфликтов.

Хабы (концентраторы) и коммутирующие хабы (коммутаторы) расширяют топологические, функциональные и скоростные возможности компьютерных сетей. Хаб с набором разнотипных портов позволяет объединять сегменты сетей с различными кабельными системами. К порту хаба можно подключать как отдельный узел сети, так и другой хаб или сегмент кабеля.

Повторители (репитеры) (рис. 3.15, а) усиливают сигналы, передаваемые по кабелю при его большой длине.

Для соединения локальных сетей используются следующие устройства, различающиеся между собой по назначению и возможностям.

Мосты (bridge) связывают две локальные сети. Они передают данные между сетями в пакетном виде, не производя в них никаких изменений. На рис. 3.15, б показаны три локальные сети, соединенные двумя мостами. Здесь мосты создали расширенную сеть, которая обеспечивает своим пользователям доступ к прежде недоступным ресурсам. Кроме того, мосты могут фильтровать пакеты, охраняя всю сеть от локальных потоков данных и пропуская наружу только те данные, которые предназначены для других сегментов сети.

Маршрутизатор объединяет сети с общим протоколом более эффективно, чем мост. Он позволяет, например, расщеплять большие сообщения на более мелкие части, обеспечивая тем

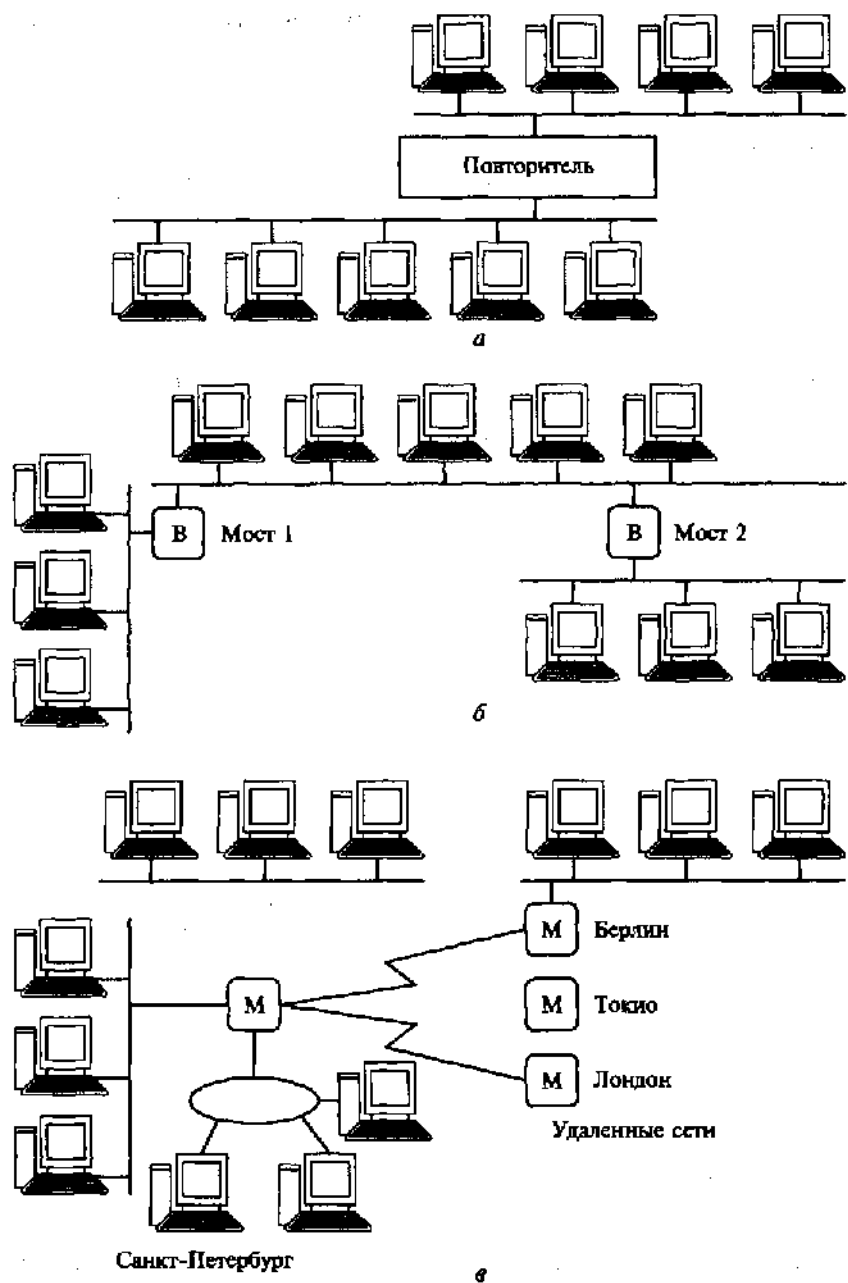


Рис. 3.15. Соединения ЛВС:
a — через повторитель; *б* — с помощью мостов (В); *в* — с помощью маршрутизатора (М)

самым взаимодействием локальных сетей с разным размером пакета. Маршрутизатор (рис. 3.15, в) может пересылать пакеты на конкретный адрес (мосты только отфильтровывают ненужные пакеты), выбирать лучший путь для прохождения пакета и многое другое. Чем сложнее и больше сеть, тем больше выгода от использования маршрутизаторов.

Мостовой маршрутизатор — это гибрид моста и маршрутизатора, который сначала пытается выполнить маршрутизацию, где это возможно, а затем в случае неудачи переходит в режим моста.

Шлюз в отличие от моста применяется в случаях, когда соединяемые сети имеют различные сетевые протоколы. Поступившее в шлюз сообщение от одной сети преобразуется в другое сообщение, соответствующее требованиям следующей сети.

Шлюзы не просто соединяют сети, а позволяют им работать как единая сеть. С помощью шлюзов также локальные сети подсоединяются к мэйнфреймам — универсальным мощным компьютерам.

Беспроводные сети (рис. 3.16) используются там, где прокладка кабелей затруднена, нецелесообразна или просто невозможна (например, в исторических зданиях, промышленных помещениях с металлическим или железобетонным полом, в офисах, полученных в краткосрочную аренду, на складах, выставках, конференциях). В этих случаях сеть реализуется при помощи сетевых радиоадаптеров, снабженных всенаправленными антеннами и использующих в качестве среды передачи информации радиоволны. Такая сеть реализуется топологией «все со всеми» и работоспособна при дальности 50...200 м (см. рис. 3.16, а).

Для связи между беспроводной и кабельной частями сети используется специальное устройство, называемое точкой входа (или радиомостом). Можно использовать и обычный компьютер, в котором установлены два сетевых адаптера — беспроводной и кабельный.

Другой важной областью применения беспроводных сетей является организация связи между удаленными сегментами локальных сетей при отсутствии инфраструктуры передачи данных (кабельных сетей общего доступа, высококачественных телефонных линий и др.), что типично для нашей страны. В этом случае для наведения беспроводных мостов между двумя удаленными сегментами используются радиомосты с антенной направленного типа (см. рис. 3.16, б).

Если в сеть нужно объединить несколько сегментов, то используется топология «Звезда» (см. рис. 3.16, в). При этом в центральном узле устанавливается всенаправленная антенна, а в удаленных узлах — направленные. Сети звездообразной топологии могут образовывать сети разнообразной конфигурации.

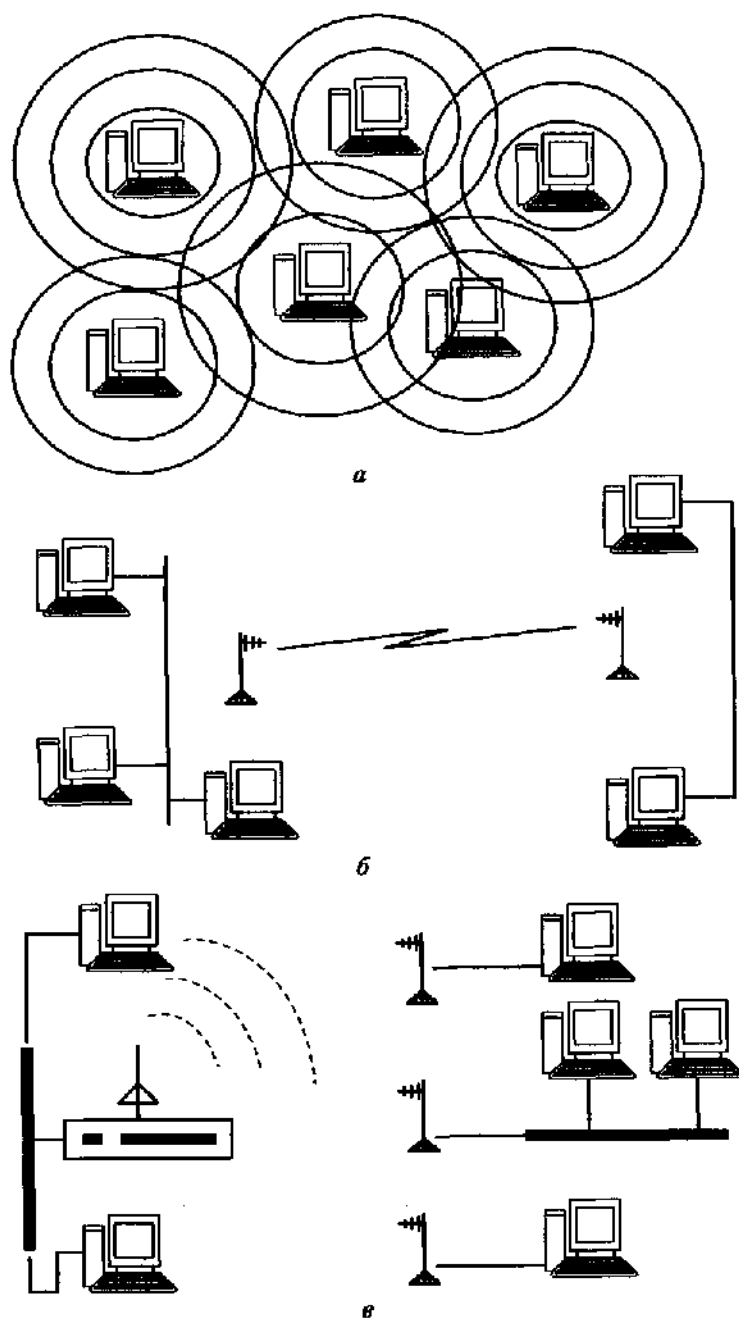


Рис. 3.16. Беспроводные сети:
a — полностью связанная технология; *b* — топология «Точка-точка»; *v* — топология «Звезда»

Сетевая магистраль с беспроводным доступом позволяет отказаться от использования медленных модемов.

3.2.7. Глобальная сеть Интернет

Интернет — гигантская всемирная компьютерная сеть, объединяющая десятки тысяч сетей всего мира. Ее назначение — обеспечить любому желающему постоянный доступ к любой информации. Интернет предлагает практически неограниченные информационные ресурсы, полезные сведения, обучающие программы, развлечения, возможность общения с компетентными людьми, услуги удаленного доступа, передачи файлов, электронной почты и многое другое. Интернет обеспечивает принципиально новый способ общения людей, не имеющий аналогов в мире.

Благодаря сети Интернет стал доступен (бесплатно или за умеренную плату) огромный объем информации. Так, пользователь в любой стране может связаться с людьми, разделяющими его интересы, или получить ценные сведения в электронных библиотеках, даже если они находятся на другом конце света. Нужная информация окажется в его компьютере за считанные секунды, пройдя путь по длинной цепочке промежуточных компьютеров, по кабелям и радиоканалам.

Интернет финансируется правительствами, научными и образовательными учреждениями, коммерческими структурами и миллионами частных лиц во всех частях света, но никто конкретно не является ее владельцем. Управляет сетью Совет по архитектуре Интернета, формируемый из приглашенных добровольцев.

Сеть была создана в 1984 г., и сейчас ею пользуются примерно 40 млн человек. Интернет все время изменяется, поскольку имеет много квалифицированных пользователей, которые пишут программы для себя, а затем распространяют их среди желающих. Постоянно появляются новые серверы, а существующие обновляются. Стремительно растут информационные потоки. Отдельные участки Интернета представляют собой сети различной архитектуры, которые связываются между собой с помощью маршрутизаторов. Передаваемые данные разбиваются на небольшие порции, называемые пакетами, которые перемещаются по сети независимо.

Сети в Интернете неограниченно коммутируются (т. е. связываются) друг с другом, потому что все компьютеры, участвующие в передаче данных, используют единый протокол коммуникации TCP/IP (ти-си-пи/ай-пи).

Протокол — правила и последовательность выполнения действий при обмене информацией на различных уровнях.

На самом деле протокол TCP/IP — это два разных протокола, определяющих различные аспекты передачи данных в сети:

- протокол TCP (Transmission Control Protocol) — протокол управления передачей данных, использующий автоматическую повторную передачу пакетов, содержащих ошибки; этот протокол отвечает за разбиение передаваемой информации на пакеты и правильное восстановление информации из пакетов получателя;

- протокол IP (Internet Protocol) — протокол межсетевого взаимодействия, отвечающий за адресацию и позволяющий пакету на пути к конечному пункту назначения проходить по нескольким сетям.

Схема передачи информации по протоколу TCP/IP такова: протокол TCP разбивает информацию на пакеты и нумерует все пакеты; затем с помощью протокола IP все пакеты передаются получателю, где с помощью протокола TCP проверяется, все ли пакеты получены; после получения всех пакетов протокол TCP располагает их в нужном порядке и собирает в единое целое.

Каждый компьютер, подключенный к сети Интернет, имеет два равноценных уникальных адреса: цифровой IP-адрес и символический доменный адрес. Присваивание адресов происходит по следующей схеме: международная организация Сетевой Информационный Центр выдает группы адресов владельцам локальных сетей, а последние распределяют конкретные адреса по своему усмотрению.

IP-адрес компьютера имеет длину 4 байт. Обычно первый и второй байты определяют адрес сети, третий байт определяет адрес подсети, а четвертый байт — адрес компьютера в подсети. Для удобства IP-адрес записывают в виде четырех чисел со значениями от 0 до 255, разделенных точками, например: 145.37.5.150 (адрес сети — 145.37; адрес подсети — 5; адрес компьютера в подсети — 150). Не могут быть использованы следующие комбинации: 255, 255.255, 255.255.255, 000, 000.000, 000.000.000, 127.

Ограничено ли число IP-адресов? Да, ограничено. В скором времени будет введена система 64-битового адреса, а пока существует проблема с распределением адресов. Провайдеры, предоставляя доступ в Интернет, выделяют компьютеру динамический IP-адрес, который действует только на время сеанса связи с Интернетом. Доменный адрес, или домен (от *англ.* domain — область), в отличие от цифрового является символическим и легче запоминается. Пример доменного адреса: t1.book.com.ru. Здесь домен t1 — имя реального компьютера, домен book — имя группы, присвоившей имя этому компьютеру, домен com — имя более крупной группы, присвоившей имя домену и т.д. В процессе передачи данных доменный адрес автоматически преобразуется в IP-адрес.

Для сопоставления IP-адресов и доменных имен, хранения и поиска адресов компьютеров в Интернете создана специальная

Контрольные вопросы

1. Опишите технологию клиент — сервер.
2. Как решается проблема совместимости интерфейсов в компьютерных сетях?
3. Дайте характеристику аппаратных средств построения сети.
4. Чем отличается архитектура сети от топологии?
5. В каких областях человеческой деятельности применяются компьютерные сети?
6. Перечислите меры защиты информации в компьютерных сетях.
7. Что общего в понятиях «архитектура компьютера» (см. подразд. 2.3) и «архитектура сети»?
8. Определите общее число IP-адресов.
9. Для чего используется доменное имя?
10. Какие функции выполняет служба безопасности компьютерной сети?

3.3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

Программное обеспечение — неотъемлемая часть компьютерной системы. Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых на компьютерах программах. Программное обеспечение (ПО) современных компьютеров включает в себя миллионы программ — от игровых до научных.

В отличие от телефона, магнитофона или телевизора, осуществляющих только заранее заложенные в них функции, персональные компьютеры могут выполнять различные действия по обработке информации. Выражение «Компьютер сделал (подсчитал, нарисовал)» означает, что на компьютере была выполнена соответствующая программа.

Под *программным обеспечением (Software)* понимается совокупность программ, которые могут выполняться вычислительной системой.

К программному обеспечению относится также вся область деятельности по проектированию и разработке ПО:

- технология проектирования программ (например, нисходящее проектирование, структурное программирование и др.);
- методы тестирования программ;
- методы доказательства правильности программ;
- анализ качества работы программ, документирование программ;
- разработка и использование программных средств, облегчающих процесс проектирования программного обеспечения, и т. д.

Для обозначения программных средств, под которыми понимается совокупность всех программ, используемых компьютера-

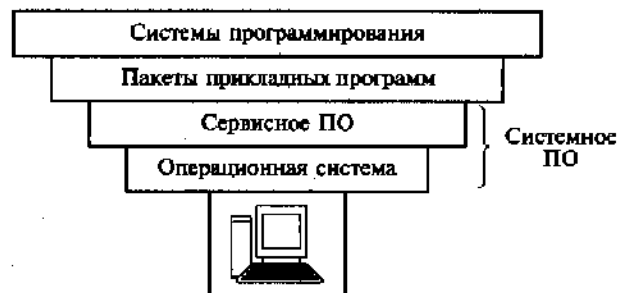


Рис. 3.17. Структура ПО персонального компьютера

ми, и область деятельности по их созданию и применению, используется термин «Software», который подчеркивает равнозначность самой машины и программного обеспечения, а также способность программного обеспечения модифицироваться, приспосабливаться и развиваться.

Базовые технические характеристики компьютера (скорость процессора, объем оперативной и видеопамати, объем жесткого диска) определяет, какое ПО будет (или может быть) установлено на нем.

Все программы можно разделить на три категории (рис. 3.17):

- системное программное обеспечение;
- пакеты прикладных программ;
- системы программирования.

3.3.1. Системное программное обеспечение

Данный класс программных продуктов тесно связан с типом компьютера и является его неотъемлемой частью.

Системное программное обеспечение направлено:

- на создание операционной среды функционирования других программ;
- обеспечение надежной и эффективной работы самого компьютера и компьютерной сети;
- проведение диагностики и профилактики аппаратуры компьютера и компьютерных сетей;
- выполнение вспомогательных технологических процессов (копирование, архивирование, восстановление файлов программ и баз данных и т.д.).

Системное ПО ориентировано на квалифицированных пользователей — профессионалов в компьютерной области: системного программиста, администратора сети, прикладного программиста, оператора.

Однако знание базовой технологии работы с этим классом программных продуктов требуется и рядовым пользователям персонального компьютера.

Программы данного класса носят общий характер применения, независимо от специфики предметной области. К ним предъявляются высокие требования по надежности и технологичности работы, удобству и эффективности использования.

Системное программное обеспечение (System Software) — совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ.

Среди десятков тысяч системных программ особое место занимают операционные системы, которые обеспечивают управление ресурсами компьютера с целью их эффективного использования (см. подразд. 3.3.4).

Важными классами системных программ являются также программы вспомогательного назначения — сервисное ПО, или утилиты (от *лат. utilitas* — польза), которые либо расширяют и дополняют соответствующие возможности операционной системы, либо решают самостоятельные важные задачи, такие как:

- программы контроля, тестирования и диагностики, которые используются для проверки правильности функционирования устройств компьютера и обнаружения неисправностей в процессе эксплуатации, указывают причину и место неисправности;
- программы-драйверы, которые расширяют возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д. С помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся;
- программы-упаковщики (архиваторы), которые позволяют записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл;
- антивирусные программы, которые предназначены для предотвращения заражения компьютерными вирусами и ликвидации последствий заражения вирусами;
- программы оптимизации и контроля качества дискового пространства;
- программы восстановления информации, форматирования, защиты данных;
- коммуникационные программы, которые организуют обмен информацией между компьютерами;
- программы для управления памятью, которые обеспечивают более гибкое использование оперативной памяти;
- программы для записи CD-ROM, CD-RW и т.д.

Часть утилит входит в состав операционной системы, а другая часть функционирует независимо от нее, т.е. автономно.

3.3.2. Пакеты прикладных программ

Пакеты прикладных программ (ППП) служат программным инструментарием решения прикладных задач и являются самым многочисленным классом программных продуктов. В данный класс входят программные продукты, выполняющие обработку информации различных предметных областей.

Установка программных продуктов на компьютер выполняется квалифицированными пользователями, а непосредственную их эксплуатацию осуществляют, как правило, конечные пользователи — потребители информации, деятельность которых во многих случаях далека от компьютерной области.

Пакет прикладных программ — комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области.

Выделим несколько классов ППП.

ППП автоматизированного проектирования. Программы этого класса предназначены для поддержки работы конструкторов и технологов, связанных с разработкой чертежей, схем, диаграмм, графическим моделированием и конструированием, созданием библиотеки стандартных элементов (темплетов), чертежей и их многократным использованием, созданием демонстрационных иллюстраций и мультфильмов.

Отличительной особенностью этого класса программных продуктов являются высокие требования к технической части системы обработки данных, наличие библиотек встроенных функций, объектов, интерфейсов с графическими системами и базами данных.

ППП общего назначения. Данный класс содержит широкий перечень программных продуктов, поддерживающих преимущественно информационные технологии конечных пользователей. Кроме конечных пользователей этими программными продуктами за счет встроенных средств технологии программирования могут пользоваться и программисты для создания усложненных программ обработки данных (см. гл. 5).

Офисные ППП. Данный класс программных продуктов включает в себя программы, обеспечивающие организационное управление деятельностью офиса: органайзеры или планировщики рабочего времени, программы-переводчики, средства распознавания текста, электронная почта и др.

Программные средства мультимедиа. Программные средства мультимедиа заняли лидирующее положение на рынке в сфере библиотечного информационного обслуживания, процессе обучения, организации досуга. Базы данных компьютерных изображений произведений искусства, библиотеки звукозаписей будут составлять основу для прикладных обучающих систем, компьютерных игр, библиотечных каталогов и фондов.

Системы искусственного интеллекта. Данный класс программных продуктов реализует отдельные функции интеллекта человека. Основными компонентами систем искусственного интеллекта являются база знаний, интеллектуальный интерфейс с пользователем и программа формирования логических выводов.

Проблемно-ориентированные ППП. Данный класс включает в себя программные продукты, обеспечивающие оперативную обработку больших объемов информации и реализующие функции управления. К этому классу относятся ППП автоматизации бухгалтерского учета, финансовой деятельности, управления кадрами и производственными ресурсами и др.

3.3.3. Системы программирования

Системы программирования обеспечивают процесс разработки программ и включают в себя специализированные программные продукты, которые являются инструментальными средствами разработчика. Программные продукты данного класса поддерживают все технологические этапы процесса проектирования, программирования (кодирования), отладки и тестирования создаваемых программ.

Системы программирования — совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов.

Эти средства на рынке программных продуктов наиболее представительны. В последнее время получили широкое распространение системы, ориентированные на создание Windows-приложений:

- пакет Borland Delphi — блестящий наследник семейства компиляторов Borland Pascal, предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования;
- пакет Microsoft Visual Basic — удобный и популярный инструмент для создания Windows-программ с использованием визуальных средств. Он содержит инструментарий для создания диаграмм и презентаций;
- пакет Borland C++ — одно из самых распространенных средств для разработки DOS- и Windows-приложений.

Перечисленные системы программирования предоставляют пользователям мощные и удобные средства разработки программ:

- транслятор (используется для перевода программ в машинные коды);
- интегрированную среду разработки;
- средства создания и редактирования текстов программ;

- обширные библиотеки стандартных программ и функций;
- отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
- «дружественную» к пользователю диалоговую среду;
- многооконный режим работы;
- мощные графические библиотеки;
- утилиты для работы с библиотеками;
- встроенный ассемблер;
- встроенную справочную службу.

Более подробно процессы программирования и этапы разработки программ описаны в гл. 6, 7, 8.

3.3.4. Операционная система

Из схемы (см. рис. 3.17) видно, что операционная система (ОС) занимает центральное место в системе программного обеспечения, выполняя связующую функцию между аппаратными средствами компьютера и прикладным или инструментальным ПО.

Компьютер выполняет действия в соответствии с предписаниями программы, созданной на одном из языков программирования. При работе пользователя на компьютере часто возникает необходимость выполнить операции с прикладной программой в целом, организовать работу внешних устройств, проверить работу различных блоков, скопировать информацию и т.п. Наиболее часто выполняемыми функциями являются:

- запуск программы на выполнение;
- организация записи программы на диск и считывание ее с диска;
- печать текста и результатов решения;
- копирование программы на другой диск;
- удаление программы с диска;
- просмотр содержимого диска и другие операции.

Эти операции используются для работы с любой программой, воспринимаемой как единое целое. Поэтому целесообразно из всего многообразия операций, выполняемых компьютером, выделить типовые и реализовать их с помощью специализированных программ.

Программы, организующие работу устройств и не связанные со спецификой решаемой задачи, вошли в состав комплекса программ, названного операционной системой.

Операционная система — совокупность программных средств, обеспечивающая управление аппаратной частью компьютера, прикладными программами, а также их взаимодействие между собой и пользователем.

Операционная система образует автономную среду, не связанную ни с одним из языков программирования. Любая прикладная программа связана с операционной системой и может эксплуатироваться только на тех компьютерах, на которых имеется аналогичная системная среда. Прикладные программные средства, разработанные в среде одной ОС, часто не могут быть использованы для работы в среде другой ОС, если нет специального комплекса программ (конвертера), позволяющего это сделать. В таком случае говорят о программной несовместимости компьютеров.

Программа, созданная в среде одной ОС, не функционирует в среде другой ОС, если в ней не обеспечена возможность конвертации (преобразования) программ.

Классифицируют ОС по числу обрабатываемых задач, числу пользователей и типу интерфейса.

По числу обрабатываемых задач ОС подразделяют на однозадачные и многозадачные.

Задачей называется программа, размещенная в оперативной памяти и обрабатываемая процессором в текущий момент времени.

Физически один процессор может выполнять только одну задачу. Многозадачность обеспечивается за счет разделения времени работы процессора на небольшие временные отрезки (кванты), предоставляемые последовательно каждой задаче. За счет высокой скорости работы процессора переключение между задачами бывает практически незаметным для пользователя, однако если запускать много задач или мощность процессора будет недостаточной, то может проявиться эффект торможения. Реальную многозадачность могут обеспечить многопроцессорные системы, которые в настоящее время широко применяются в качестве серверов.

По числу пользователей ОС подразделяют на однопользовательские и многопользовательские, или сетевые.

Многопользовательские, или сетевые, ОС включают в себя набор управляющих и обслуживающих программ, обеспечивающих обмен информацией между программами на различных ПК, доступ к общему сетевому ресурсу (например, принтеру), передачу текстовых сообщений (электронная почта) и т.д. (см. подразд. 5.5).

По типу интерфейса ОС бывают с интерфейсом командной строки и графическим интерфейсом.

Современные ОС поддерживают командный и WIMP-интерфейсы. Командный интерфейс означает выдачу на экран системного приглашения для ввода команды. Например, в MS DOS это приглашение выглядит как C:\>, в Unix — \$.

WIMP-интерфейс (от *англ.* Windows — окна, Image — образ, Menu — меню, Pointer — указатель) является графическим, т.е. на экране высвечивается окно, содержащее образы программ и меню действий. Для выбора одного из них используется указатель.

В настоящее время идут разработки речевого интерфейса SILK (от *англ.* Speech — речь, Image — образ, Language — язык, Knowledge — знание), т.е. на экране по речевой команде происходит перемещение от одних поисковых образов к другим. Предполагается, что при использовании SILK-интерфейса не нужно будет разбираться в меню.

Экранные образы достаточно точно укажут дальнейший путь перемещения от одних поисковых образов к другим по смысловым семантическим связям.

Для IBM-совместимых персональных компьютеров разработано несколько различных семейств операционных систем: MS DOS, Windows, OS/2, Unix и некоторые другие. Наиболее простой операционной системой считается однопользовательская и однозадачная операционная система MS DOS. Системы Windows, OS/2 и Unix более сложны из-за их многозадачности, а также из-за включенных в них сетевых возможностей. Первая версия операционной системы MS DOS была разработана в 1981—1982-х гг. Как было отмечено ранее, за годы существования MS DOS разработано большое число версий и модификаций этой системы. Последней версией была MS DOS 6.22. Затем была разработана значительно более мощная и удобная в использовании операционная система Windows, для которой были выпущены версии Windows 95, Windows 98 и Windows Me. Для указанных версий операционных систем часто используют одно общее обозначение Windows 9.x. Название Windows имеют и сетевые операционные системы Windows NT и Windows 2000, Windows XP. Заметим, что MS DOS оказалась как бы поглощенной, включенной в состав операционных систем Windows 9.x.

Операционная система выполняет следующие функции:

- управление работой каждого блока персонального компьютера и их взаимодействием;
- управление выполнением программ;
- организация хранения информации во внешней памяти;
- взаимодействие пользователя с компьютером, т.е. поддержка интерфейса пользователя.

Обычно операционная система хранится на жестком диске или CD-диске. При включении компьютера операционная система автоматически загружается с диска в оперативную память и занимает в ней определенное место. Этот процесс называется загрузкой операционной системы.

Рассмотрим более подробно функции операционной системы.

Управление работой периферийных устройств компьютера. Осуществляется специальными программами — драйверами.

Драйвер устройства — программа операционной системы для управления периферийными устройствами: дисководом, дисплеем, клавиатурой, принтером, манипулятором «мышь» и др.

Драйвер устройства должен учитывать специфику работы внешнего устройства, все тонкости его функционирования, поэтому каждому устройству должен соответствовать свой драйвер. Например, после нажатия клавиши на клавиатуре соответствующий драйвер выполняет обработку сигнала, проверяет, нажималась ли управляющая клавиша или производилась ли смена регистров клавиатуры. Подобные операции являются стандартными для любой программы и никакого отношения к решаемой задаче не имеют. Драйверы выполняют только стандартные функции управления вводом-выводом. Драйверы могут быть либо стандартными, либо загружаемыми.

Стандартные (внутренние) драйверы — это программы, которые находятся внутри BIOS или его модуля расширения и служат для управления внешними устройствами, входящими в стандартный комплект поставки персонального компьютера. Эти драйверы подключаются к системе автоматически после перехода компьютера в нормальное рабочее состояние. Напомним, что BIOS — это базовая система ввода-вывода, которая хранится в ПЗУ компьютера.

Загружаемые (внешние, устанавливаемые) драйверы — это программы, хранящиеся на диске и предназначенные для управления внешними устройствами, которые отличаются от стандартных либо техническими параметрами, либо режимами эксплуатации. Возможность использования загружаемых драйверов облегчает адаптацию операционной системы к новым внешним устройствам.

Управление выполнением программ. Базируется на механизме прерываний.

Прерывания — это процедуры (небольшие программы), которые компьютер вызывает для выполнения определенной задачи.

Существуют аппаратные, логические и программные прерывания.

Аппаратные прерывания инициируются аппаратурой, например сигналом от принтера, нажатием клавиши на клавиатуре, сигналом от таймера, и другими причинами. Логические прерывания возникают при нестандартных ситуациях в работе микропроцессора (например, деление на нуль, переполнение регистров и др.). Программные прерывания инициируются программами, т.е. появляются, когда одна программа хочет получить сервис со стороны другой программы (например, доступ к определенным аппаратным средствам).

Каждое прерывание имеет уникальный номер, и с ним связана определенная подпрограмма. Когда вызывается прерывание, процессор приостанавливает свою работу и выполняет прерывание: загружается адрес программы обработки прерывания и ей передается управление; после окончания ее работы управление

передается основной программе, которая была прервана. Аппаратные прерывания относятся к прерываниям низшего уровня, им присвоены младшие номера, обслуживает их базовая система ввода-вывода. Логические и программные прерывания относят к верхнему уровню, они имеют большие номера, и их обслуживает в основном базовый модуль.

Организация хранения информации во внешней памяти. Осуществляется через файловую систему. Несмотря на то что внешняя память может быть технически реализована на разных носителях (например, гибкий магнитный диск или CD-диск), их объединяет принятый в операционной системе принцип организации хранения логически связанных наборов информации в виде файлов.

Файл — это именованная совокупность данных, размещенная на внешнем запоминающем устройстве, хранимая, пересылаемая и обрабатываемая как единое целое.

Файл может содержать программу, числовые данные, текст, закодированное изображение и др.

Файл всегда имеет имя, которое состоит из двух частей, разделенных точкой: собственно имя файла и расширение, определяющее его тип (программа, числовые или текстовые данные и т.д.) (табл. 3.1). Имя файлу дает пользователь, а тип файла обычно задается программой автоматически при его создании. В различных операционных системах существуют различные форматы имен файлов. В операционной системе MS-DOS собственно имя файла должно содержать не более восьми букв латинского алфавита и цифр, а расширение состоит из трех латинских букв, например: prog1.txt. В операционной системе Windows имя файла может иметь до 255 символов, причем можно использовать русский алфавит, например: Единицы измерения информации.doc.

На каждом носителе информации может храниться большое число файлов. Порядок хранения файлов на диске определяется используемой файловой системой, которая кроме самих файлов

Таблица 3.1

Типы файлов и расширения

Типы файлов	Расширения
Программы	.exe, .com
Текстовые файлы	.txt, .doc
Графические файлы	.bmp, .gif, .jpg
Звуковые файлы	.wav, .mid, .mp3
Видеофайлы	.avi, .mov
Программы на языках программирования	.bas, .pas

включает в себя корневой каталог, систему подкаталогов, таблицу размещения файлов и называется *логической структурой диска*.

Часто возникает ситуация, когда надо работать не с одним файлом, а с группой файлов, например:

- копирование группы файлов с одного диска на другой;
- удаление группы файлов;
- перемещение группы файлов на другой диск;
- поиск группы файлов заданного типа и т. п.

Эти операции достаточно легко выполнить, пользуясь при формировании имен и типов файлов шаблоном. Шаблон имени файла — специальная форма, в которой в полях имени и типа файла используются символы «*» или «?».

Символ «*» служит для замены любой последовательности символов. В шаблоне может быть использовано в поле имени типа по одному символу «*». Например, задав имя *.TXT, вы обратитесь ко всем текстовым файлам, а задав имя SD*.8, вы обратитесь ко всем файлам, имя которых начинается на SD.

Символ «?» служит для замены одного символа. В шаблоне может быть использовано несколько таких символов. Например, имя RT??.BAS позволит обратиться ко всем файлам типа BAS, имя которых состоит из четырех символов, причем первые два символа обязательно RT, третий и четвертый — любые.

Взаимодействие пользователя с ПК. Оно обеспечивается специальными программными модулями, поддерживающими тот или иной тип интерфейса, и командным процессором, который запрашивает у пользователя команды и выполняет их.

В операционных системах с графическим интерфейсом пользователь может вводить команды с помощью мыши, тогда как в режиме командной строки необходимо вводить команды с помощью клавиатуры.

При организации хранения данных на дисках важно обеспечить быстрый доступ к данным.

3.3.5. Логическая структура диска

Логическая структура гибких дисков. Логическая структура магнитного диска представляет собой совокупность секторов (емкостью 512 байт), каждый из которых имеет свой порядковый номер (например, 100). Сектора нумеруются в линейной последовательности от первого сектора нулевой дорожки до последнего сектора последней дорожки (табл. 3.2).

Сектор является минимальным адресуемым элементом. При записи файлов на диск будет занято всегда целое число секторов, соответственно минимальный размер файла составляет один сек-

тор, а максимальный соответствует общему числу секторов на диске.

Файл записывается в произвольные свободные сектора, которые могут находиться на различных дорожках. Например, Файл_1 объемом 2 Кбайт может занимать сектора 34, 35 и 47, 48, а Файл_2 объемом 1 Кбайт — сектора 36 и 49.

Таблица 3.2

Логическая структура гибкого диска формата 3.5"

№ дорожки	№ сектора																	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
2	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
.....																		
79																		2880

Для того чтобы можно было найти файл по его имени, на диске имеется каталог. Запись о файле (табл. 3.3) содержит имя файла, адрес первого сектора, с которого начинается файл, объем файла, а также дату и время его создания.

Таблица 3.3

Структура записей в каталоге

Имя файла	Адрес первого сектора	Объем файла, Кбайт	Дата создания	Время создания
Файл_1	34	2	04.01.03	14.29
Файл_2	36	1	04.01.03	14.45

Полная информация о секторах, которые занимают файлы, содержится в таблице размещения файлов (FAT — File Allocation Table). Число ячеек FAT соответствует числу секторов на диске, а значениями ячеек являются цепочки размещения файлов, т. е. последовательности адресов секторов, в которых хранятся файлы. Например, для двух рассмотренных выше файлов таблица FAT с 1-го по 54-й сектор будет иметь вид, представленный в табл. 3.4.

При необходимости доступа к файлу сначала производится обращение к ячейке FAT-таблицы, адрес которой определяется первым номером, хранящимся в записи о файле. В этой ячейке хранится номер второго кластера этого файла. Обратившись к ячейке таблицы, соответствующей номеру второго кластера, операционная система найдет там номер третьего кластера и т. д. Так будет выстроена цепочка кластеров, в которых записан файл.

Таблица 3.4

Фрагмент FAT

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
															35	47	49
37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
										48	FFF	FFF					

В последней клетке таблицы, завершающей данную цепочку, должен находиться код FFF или FFFF для указания ее конца, т. е. конца файла.

Цепочка размещения для файла *Файл_1* выглядит следующим образом: в начальном 34-м секторе хранится адрес следующего сектора (35), соответственно в следующем, 35-м, хранится (47), в 47-м — (48), в 48-м — знак конца файла (FFF).

Для размещения каталога и таблицы FAT на гибком диске отводятся сектора со 2-го по 33-й. Первый сектор отводится для размещения загрузочной записи операционной системы. Сами файлы могут быть записаны, начиная с 34-го сектора.

Логическая структура диска формируется в процессе форматирования (см. подразд. 4.2).

Логическая структура жестких дисков. Логическая структура жестких дисков несколько отличается от логической структуры гибких дисков. Минимальным адресуемым элементом жесткого диска является кластер, который может включать в себя несколько секторов. Размер кластера зависит от типа используемой таблицы FAT и емкости жесткого диска.

Таблица FAT16 может адресовать $2^{16} = 65\,536$ кластеров. Для дисков большой емкости размер кластера оказывается слишком большим. Например, для диска объемом 40 Гбайт размер кластера был бы равен

$$40 \text{ Гбайт: } 65\,536 = 655\,360 \text{ байт} = 640 \text{ Кбайт.}$$

Файлам всегда выделяется целое число кластеров. Например, текстовый файл, содержащий слово «информатика», составляет всего 11 байт, но на диске этот файл будет занимать целиком кластер, т. е. 640 Кбайт дискового пространства. При размещении на жестком диске большого числа небольших по размеру файлов они будут занимать кластеры лишь частично, что приведет к потерям свободного дискового пространства. Эта проблема решается с помощью использования таблицы FAT32, в которой объем кла-

стера принят равным восьми секторам, или 4 Кбайт, для диска любого объема.

Замедление скорости обмена данными может происходить в результате фрагментации файлов. Фрагментация файлов (фрагменты файлов хранятся в различных, удаленных друг от друга кластерах) возрастает с течением времени, в процессе удаления одних файлов и записи других.

Так как на диске могут храниться сотни и тысячи файлов в сотнях тысячах кластеров, то фрагментированность файлов будет существенно замедлять доступ к ним (магнитным головкам придется постоянно перемещаться с дорожки на дорожку) и в конечном итоге это приведет к преждевременному износу жесткого диска. Рекомендуется периодически проводить дефрагментацию диска, в процессе которой файлы записываются в последовательно идущие друг за другом кластеры. Более подробно этот процесс будет рассмотрен в гл. 4.

В подавляющем большинстве случаев пользователю не приходится работать с таблицей FAT и ее копия, так как она заполняется при записи файла и анализируется при его считывании автоматически. Каталог, таблица FAT и ее копия (для обеспечения большей надежности таблица FAT, как правило, дублируется) автоматически создаются в процессе форматирования на любом дисковом носителе. Созданный автоматически каталог принято называть корневым.

В каталоге хранятся полные имена файлов, а также их характеристики: дата и время создания, объем (в байтах), специальные атрибуты. По аналогии с библиотечной системой организации каталогов полное имя файла, зарегистрированное в каталоге, будет служить шифром, по которому операционная система находит месторасположение файла на диске. Рассмотрим подробнее иерархическую структуру. Вооруженные силы государства в основном построены на вертикальных связях управления, т. е. по иерархическому принципу: армия—дивизия—полк—рота—взвод—военнослужащие. Можно сказать, что армия аналогична корневому каталогу, военнослужащие — файлам, а промежуточные подразделения — подкаталогам различных уровней. Такую же структуру имеют высшие учебные заведения (вуз—факультет—кафедра—преподаватели) и многие другие организации.

В информатике иерархическая структура встречается при изучении информационных моделей и баз данных.

Наиболее распространена иерархическая структура организации каталогов (рис. 3.18). На каждом диске всегда имеется главный (корневой) каталог. Он находится на нулевом уровне иерархической структуры и обозначается символом «\». Корневой каталог создается при форматировании диска. В корневой каталог могут входить другие каталоги и файлы, которые создаются или удаляются командами операционной системы.

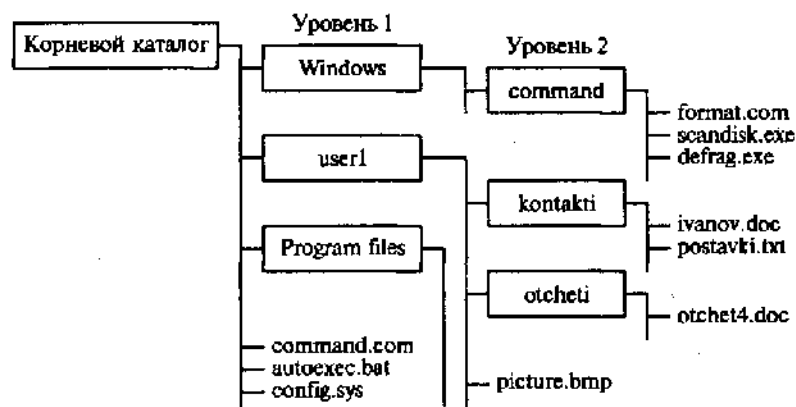


Рис. 3.18. Иерархическая структура организации каталога

Таким образом, любой каталог, содержащий каталоги нижнего уровня, может быть по отношению к ним, с одной стороны, *родительским*, а с другой стороны — *подчиненным (подкаталогом)* по отношению к каталогу верхнего уровня. Как правило, это не вызывает путаницы; употребляют термин «каталог», подразумевая или подкаталог, или родительский каталог (в зависимости от контекста). Каждый подкаталог имеет только один каталог более высокого уровня (родительский).

В некоторых операционных системах подкаталоги называются папками. Фактически подкаталоги, как и корневой каталог, являются таблицами, размещаемыми на диске и содержащими информацию об отнесенных к подкаталогу файлах. В отличие от корневого каталога положение каталогов на диске не привязано к системной области. Поэтому размеры подкаталогов могут быть достаточно произвольными, что позволяет снять ограничение на число указываемых в подкаталоге файлов.

Подкаталоги создаются пользователями по своему усмотрению. Каждый подкаталог имеет собственное имя (обычно без расширения), которое подбирается по тем же правилам, что и имя файла. Группировка и включение файлов в подкаталог могут производиться по любым критериям. Например, в отдельный подкаталог с названием «WINDOWS» целесообразно собрать все файлы, имеющие отношение к операционной системе. Точно также целесообразно сгруппировать в отдельный подкаталог все файлы, необходимые для работы какого-либо текстового редактора или игровой программы. Если на машине по очереди работают несколько пользователей, то имеет смысл организовать отдельные подкаталоги для каждого пользователя, например user1, user2, user3, ... (user — пользователь), сгруппировав в подкаталоге user1 файлы

одного пользователя, в подкаталоге user2 — второго и т.д. Кроме снятия количественных ограничений, связанных с использованием одного каталога, это создает определенную упорядоченность при хранении информации на дисках.

Операционная система осуществляет поиск файла в каталоге по его полному имени. Это означает, что в одном каталоге или подкаталоге в принципе не могут находиться два различных файла с одним и тем же полным именем. Не допускается также наличие в одном каталоге или подкаталоге двух вложенных подкаталогов с одинаковыми именами.

Для того чтобы разобраться в том, почему это невозможно, рассмотрим следующую ситуацию. Допустим, что в некотором городе почтовая служба организована так: фамилии, имена и отчества, т.е. полные имена, а также адреса всех жителей города имеются в списках, хранящихся на почте (аналог каталога на диске); отправителям корреспонденции в этот город известны только полные имена получателей, а их адреса неизвестны, поэтому на почтовых отправлениях в этот город указывается только фамилия, имя и отчество получателя. На почте по указанному в почтовом отправлении полному имени в соответствующем списке определяют адрес получателя и доставляют по нему это отправление (аналог осуществляемого операционной системой поиска файла на диске по его названию). Допустим теперь, что в этом городе проживают по разным адресам два человека с одинаковыми полными именами. Тогда рассмотренная схема работы почты приведет к тому, что адресаты с совпадающими фамилиями, именами и отчествами будут получать почтовые отправления вперемешку — и свои, и чужие.

Аналогичным образом при совпадающих полных именах двух файлов операционная система не сможет однозначно определить, в какой файл надо записывать данные или же из какого файла читать, и будет выбирать их случайным образом.

Однако в различных каталогах или подкаталогах допускается наличие файлов или дочерних подкаталогов с совпадающими полными именами. Но тогда для однозначного указания на нужный файл одного полного названия файла оказывается недостаточно. Для того чтобы отличить друг от друга файлы с одним и тем же названием, приходится указывать еще и подкаталоги, в которых они находятся. А в общем случае требуется указывать не один подкаталог, а всю цепочку подкаталогов, по которым необходимо пройти от корневого каталога до подкаталога, содержащего искомый файл, чтобы добраться до нужного файла и определить его местоположение.

Цепочка названий подкаталогов, по которым нужно пройти, начиная от корневого каталога и заканчивая подкаталогом, содержащим файл, называется *путем*, или *маршрутом*, к файлу.

В операционных системах MS DOS и Windows корневой каталог в пути указывается символом «\». Этим же символом отделяются друг от друга названия подкаталогов в цепочке, а также имя файла от названия подкаталога, в котором он находится. Таким образом, для файлов, находящихся в корневом каталоге (см. рис. 3.18), маршрутом является только обозначение корневого каталога. Файлы указываются следующим образом:

\command.com, \config.sys, \autoexec.bat

Файл из подкаталога user1 имеет путь \user1:

\user1\picture.bmp

Путь к файлам из подкаталога kontakti должен включать в себя названия обоих подкаталогов — \user1\kontakti:

\user1\kontakti\ivanov.doc, \user1\kontakti\postavki.txt

В состав компьютера, как правило, входит несколько различных дисковых устройств, поэтому для однозначного определения файла необходимо указать, на каком именно устройстве он находится. Это можно сделать, задавая название дискового устройства, содержащего файл. Полной спецификацией файла называется строка, содержащая: название устройства, путь к файлу, полное имя файла. В общем случае спецификацией называется перечисление всех отличительных особенностей. Если, например, каталог, структура которого приведена на рис. 3.18, находится на винчестере C: (каждый диск имеет логическое имя A:, B: — гибкие диски, C:, D:, E: и т.д. — жесткие и лазерные диски), то полная спецификация файла postavki.txt имеет вид

C:\user1\Kontakti\postavki.txt

В процессе работы на компьютере наиболее часто над файлами производятся следующие операции:

- копирование (копия файла помещается в другой каталог);
- перемещение (сам файл перемещается в другой каталог);
- удаление (файл удаляется из каталога);
- переименование (изменяется имя файла).

3.3.6. Загрузка операционной системы

Модули операционной системы хранятся во внешней, долговременной памяти (на жестком, гибком или лазерном диске). Однако программы могут выполняться только в том случае, если они находятся в оперативной памяти, поэтому модули операционной системы необходимо загрузить в оперативную память.

Диск, на котором находятся модули операционной системы и с которого производится ее загрузка, называется системным. Для жесткого диска (винчестера) вводится понятие «системная область» диска, так как операционная система занимает только его часть.

После включения компьютера производится загрузка операционной системы с системного диска или из системной области жесткого диска в оперативную память, которая должна выполняться в соответствии с программой загрузки.

Функции операционной системы достаточно обширны, поэтому она не может храниться, например, в ПЗУ. Однако если операционная система хранится на диске, значит, должна быть другая программа, которая поможет ее считать. Разрешение этого противоречия состоит в последовательной, поэтапной загрузке операционной системы.

Самотестирование компьютера. В состав компьютера входит энергонезависимое постоянное запоминающее устройство, содержащее программы тестирования компьютера и первого этапа загрузки операционной системы BIOS (Basic Input/Output System — базовая система ввода/вывода).

После включения питания компьютера или нажатия кнопки *Reset* на системном блоке компьютера или одновременного нажатия комбинации клавиш [Ctrl] + [Alt] + [Del] на клавиатуре процессор начинает выполнение программы самотестирования компьютера POST (Power-On Self Test). Производится тестирование работоспособности процессора, памяти и других аппаратных средств.

В процессе тестирования сначала могут выдаваться диагностические сообщения в виде различных последовательностей коротких и длинных звуковых сигналов (например: один длинный и три коротких — не подключен монитор; пять коротких — ошибка процессора и т.д.). После успешной инициализации видеокарты краткие диагностические сообщения выводятся на экран монитора.

Для установки правильной даты и времени, а также для внесения изменений в конфигурацию аппаратных средств компьютера в процессе выполнения самотестирования необходимо нажать клавишу [Del]. Загрузится системная утилита BIOS Setup, имеющая интерфейс в виде системы иерархических меню. Пользователь может установить новые параметры конфигурации компьютера и запомнить их в специальной микросхеме памяти, которая при выключенном компьютере питается от батарейки, установленной на системной плате. В случае выхода из строя батарейки конфигурационные параметры теряются и компьютер перестает нормально загружаться.

Загрузка операционной системы. После проведения самотестирования специальная программа, содержащаяся в BIOS, начинает поиск загрузчика операционной системы (рис. 3.19). Происходит поочередное обращение к имеющимся в компьютере дискам (гибким, жестким, CD-ROM) и поиск на определенном месте (в первом загрузочном секторе диска, который называется Master

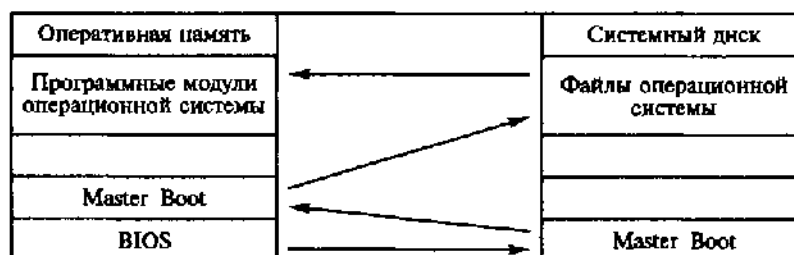


Рис. 3.19. Загрузка операционной системы

Boot, наличия специальной программы — загрузчика операционной системы).

Если диск системный и программа-загрузчик оказывается на месте, то она загружается в оперативную память и ей передается управление работой компьютера. Программа ищет файлы операционной системы на системном диске и загружает их в оперативную память в качестве программных модулей. Если системные диски в компьютере отсутствуют, то на экране монитора появляется сообщение «Non system disk» и компьютер «зависает», т.е. загрузка операционной системы прекращается и компьютер остается неработоспособным. После окончания загрузки операционной системы управление передается командному процессору. В случае использования интерфейса командной строки на экране появляется приглашение системы к вводу команд. Приглашение представляет собой последовательность символов, сообщающих о текущем диске и каталоге. Например, если загрузка операционной системы была произведена с диска C:, а операционная система была установлена в каталог WINDOWS, то появится приглашение

C:\WINDOWS>

В случае загрузки графического интерфейса операционной системы команды могут вводиться с помощью мыши. В гл. 4 рассмотрены вопросы практической работы в операционной системе WINDOWS.

3.3.7. Основные свойства и возможности Windows

Ниже перечислены характерные особенности и возможности операционной системы Windows (под названием Windows в данном учебном пособии подразумевается любая операционная система семейства Windows 9x, Windows XP).

Операционная система Windows обеспечивает единообразный интерфейс для всех своих компонентов (составных частей), а так-

же для программ, которые разработаны для работы под управлением Windows. Таким образом, освоив один раз этот интерфейс, который фактически стал стандартным для большинства прикладных программ, можно успешно работать и с незнакомыми программами.

Операционная система Windows является графической средой, поэтому управление ее работой осуществляется с помощью мыши, причем форма указателя мыши может служить определенным ориентиром в текущей ситуации.

В отличие от операционной системы MS DOS, которая может управлять выполнением только одной программы, операционная система Windows может обеспечить одновременное и независимое друг от друга выполнение нескольких программ. Все ресурсы компьютера — оперативная память, процессор, дисковые устройства, дисплей, клавиатура, вспомогательные программы — используются всеми одновременно выполняющимися программами совместно.

Операционная система своевременно и в нужном количестве выделяет каждой из этих программ нужные ресурсы, тем самым выполняя принцип многозадачности. Программа, размещенная в оперативной памяти (т.е. запущенная на выполнение), называется *задачей*.

Приложением принято называть любую прикладную или инструментальную программу, работающую под управлением Windows.

DOS-приложением называется программа, разработанная для выполнения под управлением операционной системы MS DOS. DOS-приложения отличаются тем, что все ресурсы компьютера предоставляются им в единоличное, монопольное распоряжение. Выполняться параллельно, т.е. одновременно с другими программами, они не могут.

Windows-приложения отличаются тем, что могут использовать ресурсы компьютера совместно и выполняться параллельно, т.е. одновременно с другими Windows-приложениями. Многие DOS-приложения могут выполняться и под управлением операционной системы Windows.

В то же время никакие Windows-приложения не могут выполняться под управлением MS DOS.

Непосредственно в состав операционной системы Windows включено несколько инструментальных и прикладных программ, которые обеспечивают пользователя простыми, но достаточно мощными средствами выполнения повседневных, часто встречающихся действий, таких как ввод текста, подготовка рисунка или чертежа, ведение записной книжки, рабочего ежедневника, выполнение бытовых и инженерных расчетов. К упомянутым программам относятся текстовые редакторы Блокнот и WordPad, графический редактор Paint, Адресная книга, Калькулятор, несколько

мультимедийных и игровых программ. Весь этот набор инструментов называют стандартными программами, реквизитами или аксессуарами.

Для работы под управлением Windows разработано огромное количество универсальных и специализированных инструментальных и прикладных программ. С их помощью можно создавать произвольной сложности тексты, готовить публикации, газеты, журналы, книги к изданию, создавать рисунки, тексты с рисунками, обычные и электронные таблицы, базы данных, аудио- и видеозаписи и т.д.

Все эти создаваемые с помощью инструментальных или прикладных средств объекты принято для единообразия называть документами.

Документом называется сформированный в какой-либо программе объект, содержащий некую (числовую, текстовую, графическую, звуковую, видео-) информацию.

Документ может находиться на бумажном носителе, представлять собой файл на дисковом носителе или изображение на экране дисплея.

Во время создания отдельного документа операционная система позволяет одновременно использовать несколько различных инструментальных или прикладных программ.

С их помощью в создаваемый документ могут помещаться фрагменты текста, иллюстрации, рисунки, графики, подготовленные в разных программах. Причем связывание различных фрагментов в единый документ осуществляется системой автоматически и единообразно для различных инструментальных и прикладных средств.

Операционная система обеспечивает эффективный и удобный обмен информацией между отдельными программами, выполняющимися под ее управлением. Для этого используется так называемый буфер обмена.

Буфером обмена называется участок оперативной памяти компьютера, предназначенный для временного хранения произвольного фрагмента документа, и программы, необходимой для обеспечения всех функций буфера.

Операционная система Windows кроме перечисленных выше общих возможностей обеспечивает работающим в локальной сети пользователям некоторые дополнительные возможности:

- доступ к файлам, находящимся на одном компьютере сети, пользователям, работающим на любом другом компьютере сети;
- доступ к принтеру, подключенному к одному из компьютеров сети, от любого другого компьютера сети;
- обмен сообщениями по электронной почте между пользователями;
- введение согласованного дневника планирования и выполнения заданий.

Контрольные вопросы

1. Как можно классифицировать программные продукты?
2. Что входит в системное программное обеспечение?
3. Перечислите основные функции операционной системы.
4. Что такое файл и каковы его характеристики?
5. Что входит в понятие файловой структуры диска?
6. В чем заключается процесс загрузки операционной системы?
7. Что такое Windows-приложение?
8. Как называются объекты, создаваемые с помощью инструментальных или прикладных средств?
9. Перечислите сетевые возможности Windows.

Глава 4

СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

В данной главе рассмотрены вопросы практической работы в операционной системе Windows (на примере Windows XP), раскрыты возможности наиболее распространенных сервисных программ и описаны практические приемы работы с ними.

При хранении документов не на бумажном носителе, а в электронном виде на машинных носителях информации чрезвычайно важным является вопрос обеспечения надежности хранения информации. Крайне нежелательна потеря электронного варианта документа при отсутствии его бумажного аналога. Это может привести к значительным моральным и материальным издержкам, финансовому краху предприятий, безвозвратной потере результата многолетнего труда.

Практическая работа и реальные ситуации часто требуют более широких знаний и умений сохранять в целостности информацию, защищать ее от возможных изменений, вызванных дефектами магнитных дисков, сбоями в работе ПК или воздействием компьютерных вирусов. Устранять такого рода неполадки помогут программы-утилиты, входящие в состав сервисного программного обеспечения.

4.1. РАБОТА В ОПЕРАЦИОННОЙ СИСТЕМЕ WINDOWS

Операционные системы Windows 95, Windows 98 и Windows XP компании Microsoft Corp. имеют много общего (см. подразд. 3.3.7).

Последняя версия ОС Windows — Windows XP — более комфортна и продуктивна, поддерживает современное оборудование; имеет эффективные средства диагностики, восстановления и обслуживания данных; более устойчива в работе. Наиболее важное нововведение — глубокая интеграция с Интернетом, которая, возможно, позволит увеличить срок службы операционной системы за счет наличия автоматического обновления ресурсов по каналам Интернета.

Основу работы пользователя с операционной системой Windows составляет графический интерфейс, который позволяет осуществлять взаимодействие человека с компьютером в форме диалога с использованием пиктограмм, меню, окон и т.д.

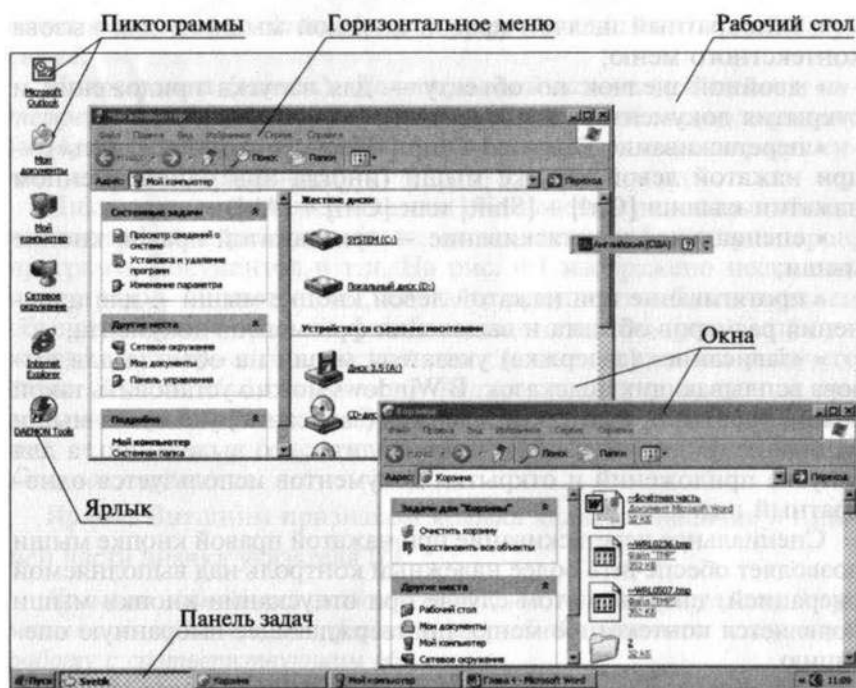


Рис. 4.1. Рабочий стол, окна и значки ОС Windows

Графический интерфейс Windows базируется на понятии «окно» (рис. 4.1), поэтому его иногда называют оконным интерфейсом. Особая роль окон в интерфейсе Windows отражена и в ее названии (от *англ.* Windows — окна). Отличительной особенностью этого интерфейса является широкое использование условных легко запоминающихся значков (пиктограмм), закрепленных за теми или иными действиями, программами, устройствами и т. д. Пользователю достаточно определенным образом указать на нужный значок, и оболочка выполнит связанное с ним действие.

4.1.1. Управление Windows с помощью мыши

Управление объектами Windows (ввод команд) осуществляется с помощью мыши и клавиатуры. Указатель (курсор) мыши в зависимости от конкретной ситуации может иметь различный вид.

Основные приемы управления с помощью мыши:

- однократный щелчок левой кнопкой мыши — для выделения и активизации объектов, ввода команд меню;

- однократный щелчок правой кнопкой мыши — для вызова контекстного меню;
- двойной щелчок по объекту — для запуска приложений и открытия документов, папок и окон;
- перетаскивание Drag-and-Drop («Переместить и оставить») — при нажатой левой кнопке мыши (иногда при одновременном нажатии клавиш [Ctrl] + [Shift] или [Ctrl] + [Alt]);
- специальное перетаскивание — при нажатой правой кнопке мыши;
- протягивание при нажатой левой кнопке мыши — для изменения размеров объекта и выделения фрагментов документа;
- «зависание» (задержка) указателя мыши на объекте для вызова всплывающих подсказок. В Windows можно установить такой режим работы, при котором задержка (зависание) указателя мыши на значке или ярлыке объекта приводит к его выделению, а для запуска приложений и открытия документов используется однократный щелчок.

Специальное перетаскивание при нажатой правой кнопке мыши позволяет обеспечить более надежный контроль над выполняемой операцией, так как в этом случае при отпускании кнопки мыши появляется контекстное меню, подтверждающее выбранную операцию.

4.1.2. Элементы интерфейса Windows

Операционная система Windows базируется на методологии объектного подхода, в соответствии с которым весь мир и любая его часть рассматривается как совокупность взаимодействующих между собой объектов. Объекты обладают определенными свойствами и поведением (различные объекты обладают различными свойствами и поведением).

Пользовательский интерфейс также построен в соответствии с принципами объектного подхода.

К объектам относятся: рабочий стол, пиктограммы, ярлык панели, меню, окна, папки, буфер обмена, приложения и документы, а также аппаратные и программные ресурсы компьютера. (Компьютер в целом тоже считается объектом.)

Рабочий стол. В операционной системе Windows весь экран дисплея рассматривается как модель рабочего места пользователя, на котором создана необходимая для обработки информации обстановка.

Рабочим столом называется элемент интерфейса пользователя, который обеспечивает эффективный доступ пользователя ко всем ресурсам компьютера, к наиболее часто используемым программам, документам и аппаратным средствам, а при наличии со-

ответствующих соединений — и к ресурсам локальной и глобальной сетей.

Пиктограмма (значок). Пиктограммой называется элемент интерфейса пользователя, представляющий собой небольшую картинку (изображение), служащую для обозначения аппаратных и программных ресурсов компьютера.

Пиктограммы используются для обозначений различных объектов интерфейса: рабочего стола, дисковых устройств, принтеров, программ, документов и т.д. На рис. 4.1 изображено несколько пиктограмм, соответствующие наиболее важным объектам интерфейса или наиболее часто используемым приложениям и документам, располагаются на поверхности рабочего стола. Значки *Мой компьютер* и *Корзина* всегда находятся на поверхности рабочего стола. Если компьютер подключен к локальной сети, то на рабочем столе обязательно присутствует значок *Сетевое окружение*.

Ярлык. Внешним признаком ярлыка является наличие в нижней части значка указателя .

Ярлыком называется значок устройства, папки, программы, документа или произвольного файла, который обеспечивает ускоренную работу с соответствующим объектом.

Ярлык содержит информацию о фактическом положении файла или папки (путь) либо команду обращения к какому-либо устройству и может располагаться не только на рабочем столе, но и в любой папке.

Панель. На рабочем столе может находиться одна или несколько панелей, которые обычно имеют вид прямоугольной вертикальной или горизонтальной полосы, окрашенной в некоторый контрастный по сравнению с рабочим столом цвет. Панель используется для размещения на ней различных значков, элементов управления и индикации состояния операционной системы Windows (см. рис. 4.1).

Панелью называется элемент интерфейса пользователя, служащий для объединения группы логически взаимосвязанных значков, элементов управления и индикации состояния операционной системы Windows.

В нижней части рабочего стола находится основная панель Windows, которую принято называть *Панелью задач*. Она играет важную роль в организации работы с окнами. Кроме того, на ней находится набор индикаторов программ, которые выполняются в так называемом фоновом режиме. Набор индикаторов имеет переменный состав. Обычно он включает в себя индикаторы текущего времени и текущего языка.

Windows является многозадачной операционной системой, т. е. одновременно могут выполняться несколько приложений. Каждое

запущенное приложение обозначается кнопкой на панели задач, при этом переход от работы одного приложения к другому может происходить с помощью щелчка по кнопке. Работающее (активное) приложение изображается на панели задач нажатой кнопкой, там же размещаются кнопки открытых в данный момент папок и кнопка *Пуск*, вызывающая *Главное меню*.

Меню. Во многих случаях, чтобы выполнить какое-либо действие, достаточно выбрать подходящий вариант из заранее подготовленного списка, который в операционной системе Windows принято называть меню.

Меню называется элемент интерфейса пользователя, представляющий собой горизонтальный или вертикальный список альтернативных вариантов команд, действий, режимов, установок, из которых пользователь должен выбрать только один вариант. Отдельные варианты, из которых состоит список, принято называть пунктами или строками меню.

По роли в управлении объектом различают следующие виды меню:

- системное;
- главное;
- подчиненное (подменю, ниспадающее меню);
- контекстное;
- пиктографическое (панели инструментов).

В Windows одна и та же операция может быть выполнена несколькими способами. Выбор способа зависит от характера выполняемой работы и периодичности ее исполнения. Каждый пользователь опытным путем подбирает наиболее удобные для себя приемы работы.

Системное меню Windows служит для управления размером, формой представления окна и для его закрытия с помощью

Таблица 4.1

Команды и кнопки управления формой представления окон в Windows

Команда Системного меню	Windows 98/NT/XP
Свернуть	
Развернуть	
Восстановить	
Закрыть	

клавиатуры. Кнопка вызова *Системного меню* находится в левой части заголовка каждого окна

Команды *Системного меню* дублируются пиктограммами (кнопками), находящимися в правой части строки заголовка окна (табл. 4.1).

Главное меню программы Windows вызывается на экран щелчком мыши по кнопке *Пуск* (рис. 4.2).

Для настройки *Главного меню* программы Windows используются, команды *Пуск/Настройка/Панель задач/Настройка меню*.

Меню окна программы обычно находится под строкой заголовка окна. Его также называют «меню окна», «горизонтальное меню», «строка меню». Горизонтальное меню окна приложения является важнейшим элементом управления, так как обеспечивает доступ ко всем функциональным возможностям программы.

Подчиненное (ниспадающее) меню выводится на экран при выборе пункта горизонтального меню. Пункты ниспадающего меню обычно называют командами.

На рис. 4.3 приведено окно документа Word 2000 с открытым (ниспадающим) меню *Файл*.

Все меню обладают общими свойствами:

- могут иметь несколько уровней;



Рис. 4.2. Главное меню

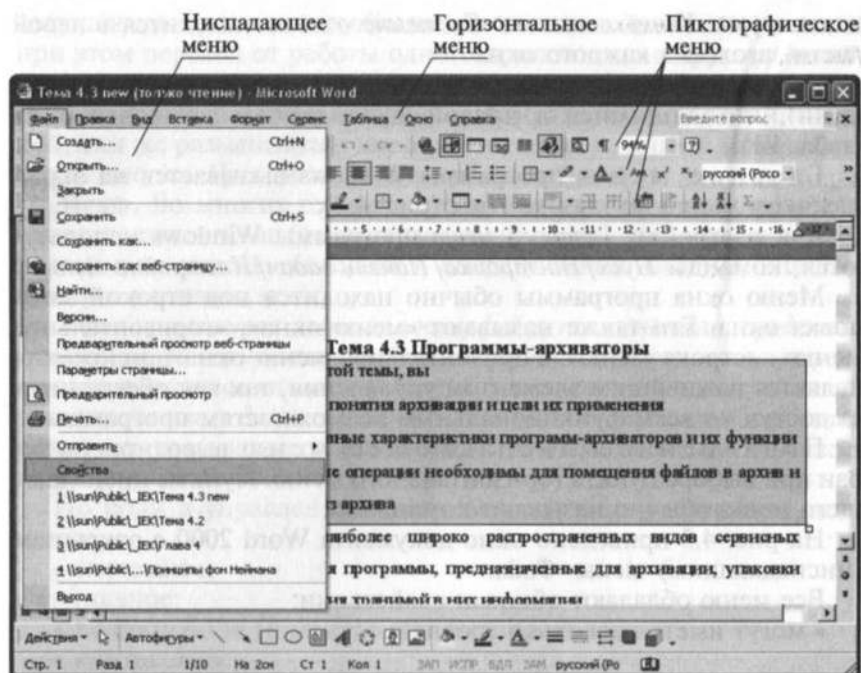


Рис. 4.3. Ниспадающее меню *Файл* в окне Word 2000, горизонтальное и пиктографическое меню

- при выборе пункта меню, имеющего после названия указатель , открывается подменю этого пункта (меню следующего уровня);

- могут иметь пункты, недоступные для выбора в данный момент (такие пункты выглядят блеклыми);

- могут иметь пункты, названия которых оканчиваются на многоточие (...); при их выборе открываются диалоговые панели.

Одна из букв в названии пункта меню, как правило, бывает подчеркнутой — это быстрая клавиша, нажатие которой приводит к открытию этого пункта меню. Например, для открытия пункта меню *Файл* достаточно нажать клавиши [Alt] + [Ф] или [Alt] + [F].

Выбрать команду ниспадающего меню можно различными способами:

- однократным щелчком мыши по пункту меню или по соответствующей пиктограмме (кнопке);

- с помощью стрелок управления курсором: нажать клавишу [F10] или [Alt] для активизации строки меню, а затем установить световое окно на пункт меню и нажать клавишу [Enter];

- нажатием «горячей» клавиши (комбинации клавиш). На рис. 4.3 комбинации клавиш указаны справа от пунктов меню: *Создать* —

[Ctrl] + [N], *Открыть* — [Ctrl] + [O], *Сохранить* — [Ctrl] + [S]; *Печать* — [Ctrl] + [P].

- нажатием «быстрой» клавиши — вначале [Alt] или [F10], а затем подчеркнутой буквы.

Закрывать меню можно нажатием клавиши [Esc] или щелчком мыши вне меню.

Контекстные меню появляются на экране после щелчка правой кнопкой мыши на объекте.

В контекстном меню отражаются операции, которые можно выполнять с данным объектом в текущей ситуации. Набор команд в каждом контекстном меню зависит от места расположения курсора мыши в момент нажатия правой кнопки.

Пиктографические меню (панели инструментов) состоят из кнопок (пиктограмм) и служат для быстрого вызова команд щелчком мыши по соответствующей кнопке (см. рис. 4.3). Панели инструментов обычно располагаются под строкой горизонтального меню окна. Их можно выводить на экран или убирать с экрана по желанию пользователя. Список всех существующих в окне панелей инструментов можно вывести на экран с помощью команды меню *Вид/Панели инструментов* или путем вызова *Контекстного меню* щелчком правой кнопки мыши по одной из находящихся на экране панелей. Щелчок на названии панели, выбранной из списка, выводит ее на экран.

Команды пиктографических меню, как правило, дублируют команды других типов меню.

В окнах приложений Microsoft Office для настройки меню и панелей инструментов используются команды *Сервис/Настройка (Вид)/Панели инструментов/Настройка*. В диалоговом окне *Настройка* на вкладках *Команды* и *Панели инструментов* можно добавлять в меню новые команды, создавать собственные панели инструментов или добавлять новые кнопки на уже существующие панели.

Окно. Каждый документ, каждый инструмент (программа, устройство) представлен на рабочем столе отдельным окном. Окна Windows являются гибкими и мощными конструкциями, обеспечивающими пользователю значительный уровень удобств во время работы. На рабочем столе одновременно может находиться произвольное число окон. Окна могут изменять свои размеры и положение на столе (накладываться друг на друга как два листа бумаги на реальном столе), даже полностью перекрывать друг друга. Например, на рис. 4.1 на поверхности рабочего стола видны два окна. Окно с названием *Мой компьютер* находится на заднем плане экрана. На него наложено окно с названием *Корзина*, которое закрывает от пользователя довольно значительную часть первого окна.

Окном называется элемент интерфейса пользователя, представляющий собой автономную часть экрана, предназначенную для орга-

низации взаимодействия между пользователем и определенной программой или некоторой ее частью.

В окне приложения выполняется любое запущенное на выполнение приложение или отражается содержимое папки. Открыть или закрыть окно приложения — то же, что и запустить программу на выполнение или завершить ее. Окна приложений можно перемещать на любое место рабочего стола, разворачивать на весь экран или сворачивать в кнопки на панели задач.

Основными элементами окна приложения являются (рис. 4.4):

- рабочая область — внутренняя часть окна, содержащая вложенные папки или окна документов;
- границы — рамка, ограничивающая окно с четырех сторон. Размеры окна можно изменять, «ухватив» и перемещая его границу мышью;
- заголовок — строка, расположенная под верхней границей окна и содержащая название окна;
- значок системного меню — кнопка, расположенная слева в строке заголовка и открывающая меню перемещения и изменения размеров окна с помощью клавиатуры;

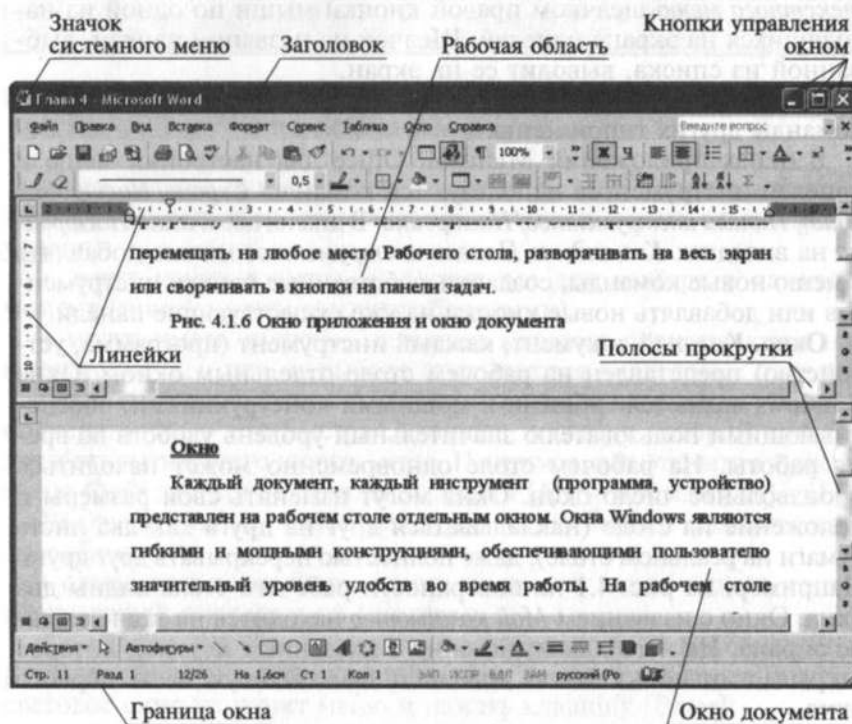


Рис. 4.4. Окно приложения и два окна документов

• строка горизонтального меню — строка, расположенная под заголовком, содержащая пункты подменю и обеспечивающая доступ к командам;

• панель инструментов — набор кнопок, располагающийся под строкой меню и обеспечивающий быстрый доступ к некоторым командам;

• кнопки *Свернуть*, *Развернуть*/*Восстановить*, *Заккрыть* — кнопки, расположенные в верхней правой части окна;

Окна документов предназначены для работы с документами и «живут» внутри окон приложений. Можно раскрывать, сворачивать, перемещать или изменять размеры этих окон, однако они всегда остаются в пределах окна своего приложения. Окно документа имеет те же кнопки управления, что и окно приложения.

Окно документа всегда содержит зону заголовка (содержащую имя документа) и часто — полосы прокрутки (появляющиеся, когда документ не помещается полностью в окне), а также линейки.

Открытое окно документа может находиться в активном либо пассивном состояниях. Если окно находится в пассивном состоянии (зона заголовка не выделена цветом), то, щелкнув по любой его части мышью, можно перевести его в активное состояние.

Диалоговое окно, или окно диалога, — окно, появляющееся на экране при вводе команды, выполнение которой требует указания дополнительных сведений. С помощью диалогового окна пользователь вводит данные, необходимые для дальнейшей работы программы.

Название диалогового окна обычно совпадает с названием команды, которая его открывает. Диалоговое окно можно перемещать по экрану обычным способом, но менять его размеры нельзя.

Диалоговые окна, как и другие окна Windows, содержат строку заголовка, иногда — строку меню, кнопки, полосы прокрутки, а также элементы, характерные только для диалоговых окон. Графические элементы диалогового окна (флажки, командные кнопки, поля ввода и т. п.) называют элементами управления.

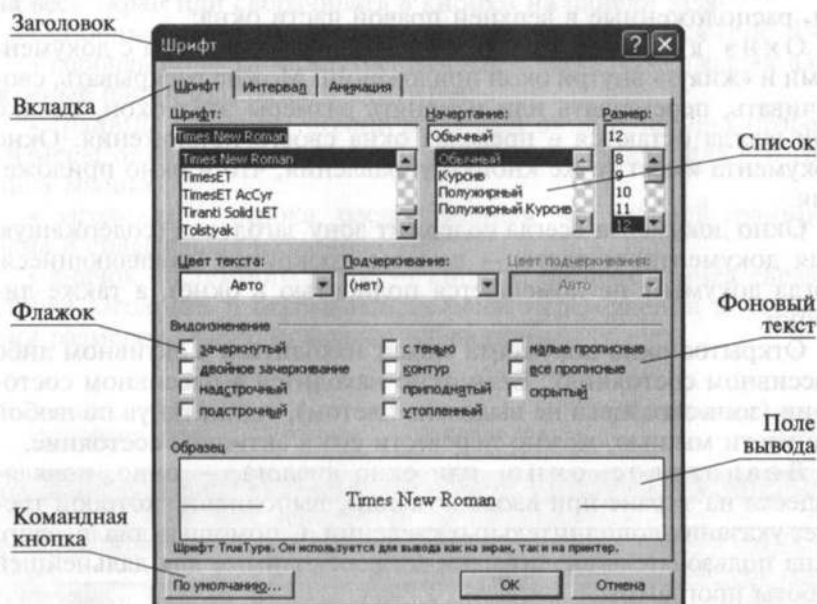
Многие диалоговые окна содержат вкладки (карточки) — «страницы», на которых можно изменять значения различных настроек.

Вкладки (карточки) характерны для диалоговых окон с несколькими функциями. Щелкнув мышью по такой вкладке (корешку карточки), можно открыть (активизировать) скрытое под ней подокно. Например, в приведенном на рис. 4.5, а диалоговом окне *Шрифт* есть три вкладки: *Шрифт*, *Интервал* и *Анимация*, т. е. в этом окне объединены три диалоговых окна.

Демонстрационное подокно *Поле вывода* служит для наглядного отображения результатов выбора. Например, в диалоговом окне

Шрифт имеется демонстрационное подокно, в котором отражаются вид и начертание выбираемого шрифта.

Фоновый текст — поясняющая надпись для элементов управления диалогового окна (например, надписи «зачеркнутый», «с тенью» и т.п.) или надписи на командных кнопках: *По умолчанию*, *OK*, *Отмена* (см. рис. 4.5, а). Подчеркнутая буква в фоновом



а



б

Рис. 4.5. Примеры диалоговых окон:

а — окно *Шрифт*; б — окно *Громкость*

тексте обозначает «быструю» клавишу, нажатием которой можно активизировать данный элемент окна.

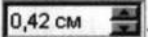
Элементы управления в диалоговых окнах:

- командные кнопки;
- поля ввода (текстовые поля);
- счетчики;
- списки;
- кнопки выбора (радиокнопки);
- переключатели (флажки);
- ползунки.

При активизации командных кнопок (щелчком мыши или с помощью клавиатуры) сразу же выполняется соответствующая команда.

Поля ввода (текстовые поля) — место для ввода текстовой информации. Это может быть дата, имя файла (при открытии, поиске или сохранении файла), параметры страницы и т.п. Для того чтобы в текстовое поле ввести информацию, необходимо сделать по нему щелчок мышью, после чего в поле появится мигающая вертикальная черта — текстовый курсор, показывающий место для ввода символов.

Существуют комбинированные поля ввода, которые имеют список и полосу прокрутки или числовые значения, которые можно менять с определенным шагом.

Счетчики представляют собой пару стрелок, которые позволяют увеличивать или уменьшать значения в связанном с ними текстовом поле: .

Наличие стрелок рядом с текстовым полем не означает, что в это поле нельзя ввести нужное значение обычным способом (с помощью клавиатуры). Во многих случаях ввести значение с помощью клавиатуры можно легче и быстрее, чем с помощью стрелок.

С п и с о к — перечень объектов для выбора (см. рис. 4.5, а). Выбор из списка осуществляется щелчком мыши по нужному элементу. В большинстве списков можно выбрать только одно значение, однако некоторые списки позволяют отметить два значения и более. В этом случае выбор из списка нужно делать при нажатой клавише [Ctrl].

Раскрывающийся список выглядит как текстовое поле, снабженное кнопкой с направленной вниз стрелкой  (*Цвет текста* и *Подчеркивание*) (см. рис. 4.5, а). Для раскрытия списка нужно щелкнуть по находящейся справа кнопке со стрелкой. Используя клавиши управления, курсором можно перемещаться по элементам раскрывающегося списка.

К н о п к и в ы б о р а (радиокнопки) служат для выбора одного из взаимоисключающих вариантов. Обычно переключатели обо-

значаются кружками. Переключение режимов осуществляется щелчком мыши. При этом в кружке активизированного переключателя появляется точка .

Переключатели (флажки) могут располагаться как группами, так и отдельно, причем каждый параметр является независимым. Их можно выбрать несколько, а можно не выбрать ни одного. Флажки обозначаются квадратиками , стоящими рядом с названием режима, функции или параметра. Для того чтобы установить (выбрать) флажок, необходимо щелкнуть по его квадратiku или по стоящей рядом текстовой строке. Чтобы сбросить флажок (отменить выбор), нужно сделать повторный щелчок. Когда флажок установлен, в квадратике стоит галочка  (см. рис. 4.5, а, б). Квадратики некоторых флажков могут быть закрашены серым цветом. Это означает, что флажок установлен лишь для части выделенных объектов.

Ползунки используются в некоторых диалоговых окнах для увеличения или уменьшения значений параметров (рис. 4.5, б).

Элемент управления в диалоговом окне (флажок, поле выбора, командная кнопка) реагирует на сигнал клавиатуры только в том случае, если этот элемент обладает фокусом (например, фокусом обладает строка, в которую можно вводить данные). Установить фокус на элементе можно щелчком мыши или с помощью клавиши [Tab].

Наличие фокуса у элемента можно определить по следующим признакам:

- в поле ввода мерцает текстовый курсор;
- командная кнопка выделена утолщенной черной рамкой;
- у остальных элементов (в списке или у фоновом тексте) можно заметить пунктирное окаймление.

Установив фокус на нужный элемент, с помощью клавиатуры можно сделать следующее:

- ввести данные (текст, цифры) в поле ввода;
- установить или сбросить флажок нажатием клавиши [Пробел];
- ввести команду нажатием клавиши [Enter];
- выбрать элемент списка или включить/выключить поле выбора — радиокнопку стрелками перемещения курсора.

Для перемещения фокуса по элементам окна вперед используется клавиша [Tab], а для перемещения фокуса по элементам окна назад — комбинация клавиш [Shift] + [Tab].

Заккрыть диалоговое окно можно нажатием клавиши [Esc] или с помощью кнопки .

После открытия диалогового окна в окне программы (приложения) дальнейшая работа может идти по-разному:

- открытое диалоговое окно может заблокировать (остановить) работу в окне приложения;

- в окне приложения можно производить работу и при открытом диалоговом окне.

В соответствии с этим диалоговые окна подразделяются на два типа: модальные и немодальные.

Модальные диалоговые окна блокируют работу приложения. Для того чтобы вернуться в окно приложения (документа) и продолжить работу, нужно завершить все операции с таким окном и закрыть его. Наличие кнопки *ОК* является признаком модального окна. Примерами модальных диалоговых окон могут служить окна *Шрифт*, *Открытие документа*, *Печать*, *Сохранение документа*.

Немодальные диалоговые окна (например, *Символ*, *Перейти*, *Найти* и *Заменить* и др.) не останавливают работу приложения. Можно, не закрывая текущее окно, переходить в окно приложения (документа), работать с ним, а затем возвращаться в диалоговое окно. В немодальных окнах кнопка *ОК*, как правило, отсутствует, зато для их закрытия имеется кнопка *Отмена* или кнопка *Закрыть*. Переключаться в документ и обратно можно щелчком мыши.

Диалоговые окна отличаются большим разнообразием. Иногда они содержат несколько вкладок. Например, в программе Word диалоговое окно *Параметры* (которое открывается командой *Сервис/Параметры* содержит 10 вкладок.

В некоторых диалоговых окнах Windows и его приложений (*Открытие документа*, *Сохранение документа*, *Найти*) можно работать, как в обычных окнах папок.

Например, найдя объект с помощью команды *Поиск/Файлы и папки*, непосредственно из диалогового окна *Найти/Все файлы* можно выполнить следующие операции:

- запустить программу или открыть документ;
- вывести документ на печать;
- просмотреть свойства файла или папки;
- поместить объект в буфер обмена;
- переименовать или удалить объект;
- создать архив или ярлык;
- просмотреть содержимое документа, не открывая его.

Для выполнения этих операций следует использовать кнопки, команды меню диалогового окна (*Файл*, *Просмотр*) или аналогичные команды контекстного меню.

Умение работать с диалоговыми окнами и правильный ввод необходимых параметров во многом определяют успех работы в целом. Информация, которая содержится на вкладках, и набор элементов управления в конкретном диалоговом окне требуют внимательного анализа и служат своеобразной подсказкой пользователю о том, какие возможности имеются в системе для выполнения той или иной команды.

В затруднительных случаях очень полезно использовать кнопку  (Справка), которая находится в строке заголовка диалогового окна.

Папка. Для повышения удобства работы с программами и документами в операционной системе Windows предусмотрен объект с названием «Папка с файлами...» или просто «Папка...». Выделенную по какому-либо признаку группу документов и/или приложений можно положить в папку и работать с этой группой как с единым целым. Здесь совершенно четко просматривается аналогия и по назначению, и по названию с обычной папкой для бумаг. Фактически папка с файлами операционной системы Windows является подкаталогом на каком-либо дисковом устройстве. В силу этого в папке могут находиться произвольные файлы, а не только файлы документов.

Папкой с файлами называется элемент интерфейса пользователя, предназначенный для группировки файлов, программ и документов, по какому-либо признаку.

Важнейшим свойством папок является их иерархическая структура (см. подразд. 3.4). Это означает, что в папке может находиться не только произвольное число файлов, но и вложенные в нее папки. Папка, в которой находится другая папка, считается родительской по отношению к любым вложенным в нее папкам. Все вложенные папки считаются дочерними по отношению к папке, в которую они вложены.

Папкам соответствует значок , похожий на закрытую папку. На некоторых значках могут быть изображены пиктограммы объектов, связанных с папкой. Например, папка, связанная с принтером, выглядит следующим образом: . В операционной системе Windows предусмотрена стандартная папка *Мои документы*. Ей соответствует значок , похожий на раскрытую папку. Предполагается, что в эту папку пользователь помещает созданные им документы, хотя это совершенно не обязательно. В папку *Мои документы* можно поместить любой документ или приложение, а свои документы пользователь может положить в любую другую папку. Тем не менее обычно пользователи стараются придерживаться правила хранения своих документов в этой папке.

В операционной системе Windows понятие «папка» имеет более широкое толкование.

В общем случае папка представляет собой объект, который обеспечивает единообразие терминологии и действий по доступу к любым ресурсам, организованным по иерархическому принципу.

Для обеспечения доступа к любым ресурсам кроме обычных папок в операционной системе Windows вводятся так называемые системные папки, с помощью которых можно, например,

обеспечить доступ к принтерам, подключенным к данному компьютеру.

Системные папки служат для доступа к любым аппаратным и программным ресурсам компьютера или сети, в том числе и к обычным папкам. Системные папки, в отличие от папок с файлами, образуются операционной системой автоматически, в то время как папки с файлами создаются и уничтожаются по специальным командам пользователя.

Буфер обмена. В Windows существует возможность обмена информацией между различными приложениями с помощью буфера обмена (см. подразд. 3.4). *Буфер обмена* — область памяти, в которую временно помещается вырезанный или скопированный объект или фрагмент документа. Управляет работой буфера обмена специальная программа.

Записанный в буфер фрагмент можно вставить либо в другое место того же документа, либо в другой документ того же приложения, либо в документ другого приложения. Фрагмент, находящийся в буфере обмена, может вставляться произвольное число раз. Объект или фрагмент сохраняется в буфере обмена Windows до тех пор, пока не поступила следующая команда поместить в буфер новую порцию данных.

Разные приложения формируют документы, имеющие разные формы представления информации. Для текстовых документов используются одни способы кодирования, а для рисунков и чертежей, создаваемых в графических редакторах, — другие. Напомним, что форма представления информации в документе, создаваемом в том или ином приложении, называется форматом документа. Формат сохраняется у всех попадающих в буфер фрагментов документа. Но фрагмент, помещенный в буфер обмена и имеющий формат исходного документа, может быть вставлен в любой другой документ, в том числе, и в документ с каким-либо другим форматом. Таким образом, во время выполнения вставки из буфера возникает необходимость преобразования исходного формата фрагмента к формату документа, в который выполняется вставка. Все необходимые преобразования при передаче фрагмента из одного документа в другой через буфер обмена операционная система берет на себя.

С помощью клавиши [PrintScreen] можно скопировать и поместить в буфер обмена графический образ всего экрана. При нажатии комбинации клавиш [Alt] + [PrintScreen] в буфер обмена копируется только активное окно.

В Windows с помощью буфера обмена можно обмениваться информацией с приложениями MS-DOS.

В Microsoft Office 2003 имеется расширенный буфер обмена, в котором одновременно могут сохраняться до 24 объектов или фрагментов. Управление работой расширенного буфера обмена осуще-

ствляется с помощью специальной панели инструментов *Буфер обмена*.

Чтобы поместить объект в буфер обмена, этот объект необходимо предварительно выделить, а затем использовать один из способов:

- ввести команду меню *Правка/Копировать* или *Правка/Вырезать*;
- щелкнуть по кнопке *Копировать* или *Вырезать* на панели инструментов;
- ввести команду контекстного меню *Копировать* или *Вырезать*;
- нажать быстрые клавиши [Ctrl] + [C] (копировать) или [Ctrl] + [X] (вырезать).

Для помещения в документ объекта из буфера обмена можно использовать:

- команды меню *Правка/Вставить* или *Правка/Специальная вставка*;
- кнопку *Вставить* на панели инструментов;
- команду контекстного меню *Вставить*;
- клавиши быстрого ввода [Ctrl] + [V] или [Shift] + [Insert] (вставить).

Для создания и совместного использования объектов можно также создавать фрагменты (OLE-объекты), которые могут храниться в папках или на рабочем столе. Кроме того, содержимое буфера обмена можно записать на постоянное хранение в файл специального формата расширением: *.clp.

Вызов команд для выполнения различных операций Windows приведен в табл. 4.2.

Таблица 4.2

Основные операции в операционной системе Windows

Действие	Команда, клавиши
<i>Сводка команд справочной системы</i>	
Вызов справки операционной системы Windows	 (Справка), [Windows] + [F1]
Вызов справки окна	Справка/Вызов справки, [F1]
Вызов контекстной справки	?/  /Что это такое?, [Shift] + [F1]
Перемещение между областями окна	[F6]
Перемещение между ссылками в окне	[Tab], [Shift] + [Tab]
Выбор ссылки	[Enter]

Продолжение табл. 4.2

Действие	Команда, клавиши
Прокрутка содержимого окна	[↓], [↑], [→], [←], [Page Up], [Page Down]
Переход к ранее просмотренному разделу	<i>Параметры/Назад</i> , [Alt] + [←]
Возврат от ранее просмотренного раздела	<i>Параметры/Вперед</i> , [Alt] + [→]
Возврат к первому разделу	<i>Параметры/Домой</i>
<i>Работа с основным меню</i>	
Открытие основного меню	<i>Windows</i> , [Ctrl] + [Esc]
Закрытие основного меню	[Esc]
Перемещение по строкам меню	[↓], [↑]
Открытие следующего уровня каскадного меню	[→]
Возврат к предыдущему уровню меню	[←], [Esc]
Выбор команды в меню	[Enter]
<i>Работа с окном Мой компьютер и окном программы Проводник</i>	
Открытие окна <i>Мой компьютер</i>	[Windows] + [D]
Вывести\убрать панель «Адрес»	<i>Вид/Панели инструментов/Адресная строка</i>
Вывести\убрать строку состояния	<i>Вид/Строка состояния</i>
Список ресурсов в виде крупных значков	<i>Вид/Крупные значки</i>
Список ресурсов в виде мелких значков	<i>Вид/Мелкие значки</i>
Список ресурсов в стандартном виде	<i>Вид/Список</i>
Список в виде таблицы	<i>Вид/Таблица</i>
Обновление содержания списка	<i>Вид/Обновить</i> , [F5]
Переход на уровень вверх	<i>Переход/На один уровень вверх</i> , [Backspace]
Переход к предыдущему состоянию	[Alt] + [←]
Возврат от предыдущего состояния	[Alt] + [→]

Продолжение табл. 4.2

Действие	Команда, клавиши
<i>Только для окна программы Проводник</i>	
Переход между областями окна	[Tab], [F6]
Развертывание уровня дерева папок	[→]
Свертывание уровня дерева папок	[←]
<i>Файловые операции</i>	
Создание папки или файла	<i>Файл/Создать</i> , [Ctrl] + [N]
Первичное сохранение файла	<i>Файл/Сохранить как...</i>
Открытие документа	<i>Файл/Открыть...</i> , [Ctrl] + [O]
Сохранение документа	<i>Файл/Сохранить</i> , [Ctrl] + [S]
Закрытие документа	<i>Файл/Заккрыть</i>
Определение свойств папки или файла	<i>Файл/Свойства</i> , [Alt] + [Enter]
Переименование папки или файла	<i>Файл/Переименовать</i> , [F2]
Выделение всего списка	<i>Правка/Выделить все</i> , [Ctrl] + [A]
Копирование группы папок и файлов	<i>Правка/Копировать</i> + <i>Правка/Вставить</i> , [Ctrl] + [C], затем [Ctrl] + [V]
Перемещение группы папок и файлов	<i>Правка/Вырезать</i> + <i>Правка/Вставить</i> , [Ctrl] + [X], затем [Ctrl] + [V]
Удаление группы папок и файлов	<i>Файл/Удалить</i> , [Delete]
Удаление группы папок и файлов без корзины	[Shift] + [Delete]
Восстановление папок и файлов из корзины	<i>Файл/Восстановить</i>
Печать документа	<i>Файл/Печать</i> , [Ctrl] + [P]
<i>Работа с буфером обмена</i>	
Перенос фрагмента в буфер обмена	<i>Правка/Вырезать</i> , [Ctrl] + [X]
Копирование фрагмента в буфер обмена	<i>Правка/Копировать</i> , [Ctrl] + [C]
Вставка фрагмента в документ из буфера обмена	<i>Правка/Вставить</i> , [Ctrl] + [V]
Копирование экрана в буфер	[PrintScreen]

Окончание табл. 4.2

Действие	Команда, клавиши
<i>Работа с ярлыками</i>	
Создание ярлыка	<i>Контекстное меню/Создать ярлык</i>
Удаление ярлыка	<i>Контекстное меню/Удалить</i>
Изменение ярлыка	<i>Контекстное меню/Переименовать</i>
<i>Работа с объектами</i>	
Выделение нескольких объектов	[Ctrl] + выделение указателем мыши
Выделение нескольких последовательно расположенных объектов	[Shift] + выделение указателем мыши
Сортировка всех объектов	<i>Выделение/Вызов контекстного меню/Упорядочить значки</i>
Удаление	Выделение указателем мыши/[Del]
<i>Работа с корзиной</i>	
Просмотреть содержимое корзины	<i>Контекстное меню</i>
Очистить корзину	<i>Контекстное меню/Открыть/Файл/Очистить корзину</i>
Изменение размера корзины	<i>Свойства/Глобальные/Объем корзины</i>

Контрольные вопросы

1. Перечислите элементы графического интерфейса.
2. Чем отличаются понятия «пиктограмма» и «ярлык»?
3. Какие типы окон используются в операционной системе Windows?
4. Назовите отличия в структуре и свойствах диалоговых окон и окон других типов.
5. Чем отличаются списки от переключателей?
6. Для чего нужен буфер обмена?
7. Можно ли создать ярлык на ярлык; если да, то какая информация будет в нем содержаться?
8. Приведите примеры задач, для которых пользователю было бы необходимо держать открытыми два или три окна.
9. В чем заключается документно-ориентированный стиль работы?

4.2. ПРОГРАММЫ ОБСЛУЖИВАНИЯ МАГНИТНЫХ ДИСКОВ

Магнитные диски, которые используются в качестве долговременных носителей информации, а также особенности существующего на сегодняшний день системного программного обеспече-

ния не гарантируют 100%-й сохранности записанной на диски информации даже при условии прекрасного физического состояния дисков. Файлы, в которых хранится информация, могут быть случайно уничтожены самим пользователем. В результате сбоев в работе программного обеспечения или аппаратуры может оказаться разрушенной файловая система и весь файл или некоторые его участки могут оказаться потерянными, недоступными для выборки информации. Существует еще десяток причин, по которым может произойти потеря ценной информации, поэтому необходимо постоянное внимание к дисковым носителям информации и контроль за состоянием файлов.

4.2.1. Форматирование дисков

Магнитные диски по мере их эксплуатации теряют свою надежность, на них появляются сбойные сектора и дорожки. Информация, записанная на такие участки диска, может быть впоследствии потеряна, ее невозможно будет прочитать, поэтому перед записью важной информации на диск его полезно переформатировать. Кроме того, новые дискеты могут оказаться еще не отформатированными, в этом случае придется выполнять их первичное форматирование. В процессе форматирования на рабочих поверхностях дискеты формируются дорожки и рабочие сектора. Кроме того, на дискете формируются необходимые таблицы файловой системы: корневой каталог, FAT и т.д.

Следует помнить о том, что в процессе выполнения форматирования вся информация, которая находилась на диске, будет уничтожена. Перед форматированием необходимо убедиться в том, что на диске не осталось полезной информации.

Вся справочная информация о числе секторов, дорожек; размере сектора; количестве рабочих поверхностей; начале области размещения данных и некоторая другая сохраняется в первом секторе нулевой дорожки диска. Этот сектор является своеобразным паспортом диска и называется *boot-сектором*. Любое повреждение boot-сектора (физическое или вирусное) влечет за собой невозможность доступа к данным.

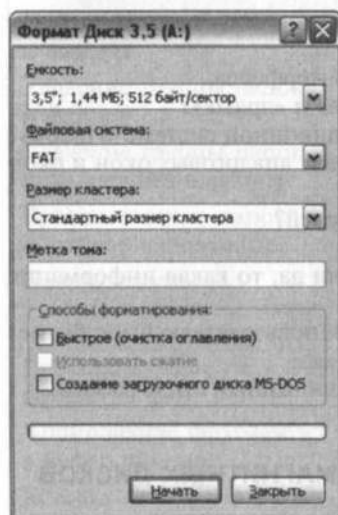


Рис. 4.6. Окно форматирования гибкого диска

Форматирование дисков выполняется с помощью команды *Файл/Форматировать* окна *Мой компьютер* или программы *Проводник*, диалоговое окно которого показано на рис. 4.6. В операционной системе Windows предусмотрены три способа форматирования дисков, которые выбираются с помощью включения соответствующего переключателя в группе *Способы форматирования* окна *Форматирование*:

1) быстрое форматирование — формируются новые таблицы файловой системы диска, физическая разметка рабочих поверхностей не производится;

2) полное форматирование — формируются новые таблицы файловой системы диска и производится физическая разметка рабочих поверхностей;

3) копирование системных файлов — новые таблицы файловой системы не создаются, физической разметки рабочих поверхностей не происходит, обновляются только основные файлы операционной системы (режим переформатирования для системных дисков).

Дискету можно отформатировать с заданием метки тома или без ее задания. Метка представляет собой название, содержащее не более 11 символов. Это название, если оно задано, появляется в строке соответствующего дискового устройства в списках окон *Мой компьютер* и *Проводник*. По результатам форматирования всегда формируется отчет, который может быть выведен или не выведен на экран после окончания форматирования. И, наконец, можно включить или выключить режим создания в процессе форматирования системного диска, т. е. запись файлов операционной системы.

Не рекомендуется самостоятельно выполнять форматирование жестких дисков, так как это может повлечь за собой потерю значительных массивов личной информации и дорогостоящего программного обеспечения.

4.2.2. Копирование гибких дисков

Копирование содержимого одного гибкого диска на другой представляет собой достаточно часто встречающуюся на практике задачу. Диск, с которого нужно снять копию, называется исходным, а диск, который должен стать копией исходного, — целевым. Задачу копирования можно решить в два приема: сначала скопировать содержимое исходного диска на жесткий диск, а затем выполнить копирование с жесткого диска на целевой.

Чтобы избежать дополнительных временных затрат и дополнительного расхода дисковой памяти, можно воспользоваться специальными средствами операционной системы Windows, кото-

рые позволяют выполнить копирование с одного гибкого диска на другой с помощью только одного дисководов и без промежуточной записи на жесткий диск (необходимо открыть исходный диск). Во время копирования с исходного диска снимается полная и точная побитовая копия (команда *Файл/Копировать диск* или команда *Копировать диск* контекстного меню). Для этого целевой диск сначала автоматически форматируется. Это означает, что все, что находится на целевом диске, во время копирования будет уничтожено.

После завершения чтения содержимого исходного диска на экран будет выведено окно с указанием «установить целевой диск». Нужно вынуть из дисководов исходный диск, вставить целевой и нажать кнопку *ОК*. Если целевой диск имеет сбойные сектора, то во время предварительного форматирования может быть выяснено, что на целевом диске не хватит места для точной копии исходного. В этом случае на экран будет выведено окно сообщения об ошибке при форматировании. Следует заменить целевую дискету и нажать кнопку *Повтор* или отказаться от копирования.

4.2.3. Определение объема свободного пространства диска

Перед выполнением записи файла на диск полезно заранее знать, поместится данный файл на данный диск или нет. Для этого нужно сравнить размер файла и объем свободного пространства на диске. Для определения объема свободного пространства на диске можно использовать как окно *Мой компьютер*, так и окно программы *Проводник*, или можно открыть окно *Свойства* контекстного меню диска и в нем найти исчерпывающую информацию о его состоянии (рис. 4.7, а).

4.2.4. Проверка рабочих поверхностей дисков

В процессе эксплуатации магнитных дисков на их рабочих поверхностях могут возникать различные дефекты. Если в секторе, размещенном на дефектном участке рабочей поверхности, хранилась какая-либо информация, то она может оказаться разрушенной или недоступной. Другая опасность состоит в том, что сектора, размещенные на дефектных участках, могут быть использованы при записи новой информации наряду с нормальными секторами, и тогда может быть потеряна не только старая информация, но и вновь записываемая.

Напомним, что вследствие различных сбоев аппаратуры, ошибок в программном обеспечении или в результате ошибочных дей-

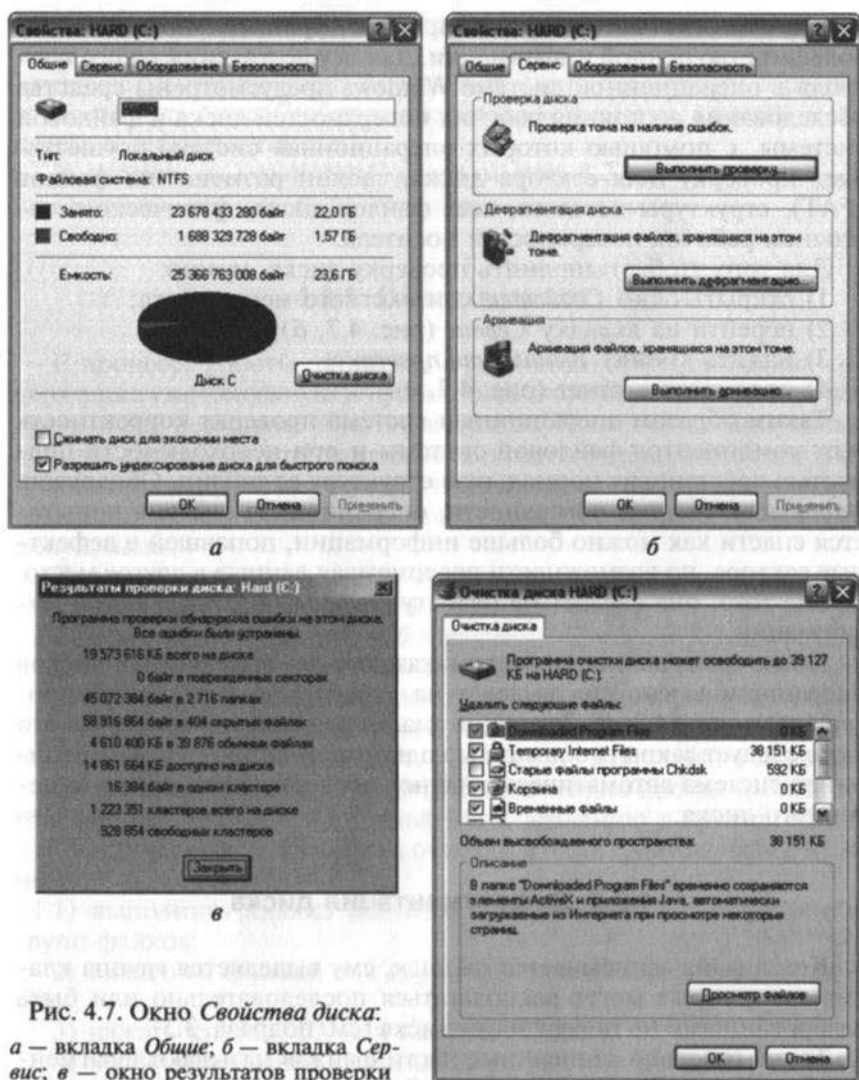


Рис. 4.7. Окно *Свойства диска*.

а — вкладка *Общие*; б — вкладка *Сервис*; в — окно результатов проверки диска; г — окно *Очистка диска*

ствий пользователя может оказаться частично поврежденной или полностью разрушенной файловая система диска (каталог таблицы размещения файлов и т. д.), что также может привести к частичной или полной потере информации, хранившейся на диске.

Для того чтобы при хранении информации на дисковых носителях избежать ее потери, необходимо периодически контролировать качество рабочих поверхностей и целостность файловой системы на всех используемых дисках. Особенно это важно для жест-

ких дисков, на которых, как правило, хранится подавляющее большинство ценной информации. Для осуществления такого контроля в операционной системе Windows предусмотрены средства обследования состояния рабочих поверхностей диска и файловой системы, с помощью которых операционная система осуществляет проверку boot-сектора диска, таблиц размещения файлов (FAT), структуры каталога, всех файлов диска, физического состояния рабочих поверхностей носителя.

Для того чтобы выполнить проверку диска, нужно:

- 1) открыть окно *Свойства* контекстного меню диска;
- 2) перейти на вкладку *Сервис* (рис. 4.7, б);
- 3) нажать кнопку *Выполнить проверку*;
- 4) просмотреть отчет (рис. 4.7, в).

Таким образом операционная система проверит корректность всех компонентов файловой системы и при необходимости полностью восстановит нормальную структуру ее таблиц. Обнаружив дефектные участки поверхности, операционная система попытается спасти как можно больше информации, попавшей в дефектные сектора, по возможности переписывая данные в другое место. Кроме того, она сделает их недоступными для записи новой информации.

После завершения проверки каждого из выделенных дисков операционная система выведет на экран окно результатов проверки (см. рис. 4.7, в). Завершив анализ результатов проверки, это окно следует закрыть с помощью одноименной кнопки. Операционная система автоматически начнет проверку следующего выделенного диска.

4.2.5. Дефрагментация диска

Когда файл записывается на диск, ему выделяется группа кластеров, которые могут располагаться последовательно или быть разбросанными по поверхности диска (см. подразд. 3.3).

Такие отдельно записанные части файлов называют фрагментами, а сам разбитый на части файл называют фрагментированным. Сведения о том, где именно находятся отдельные фрагменты файла, собраны в таблице файловой системы FAT. Если диск эксплуатируется в режиме с многочисленными удалениями и записями файлов, то файлы могут оказаться разбитыми на очень большое число частей.

Для увеличения срока службы дисководов, особенно это важно для жестких дисков, рекомендуется периодически, несколько раз в месяц, подвергать диски процедуре дефрагментации. Во время ее выполнения все файлы на диске перезаписываются так, чтобы каждый файл занимал один сплошной участок диска, и,

следовательно, размещение файлов на диске окажется оптимальным для работы компьютера.

Дефрагментацию диска можно выполнить следующим образом:

- 1) открыть окно *Свойства* дискового устройства, которое нужно в дефрагментации;
- 2) перейти на страницу *Сервис* (см. рис. 4.7, б);
- 3) нажать кнопку *Выполнить дефрагментацию*.

4.2.6. Очистка диска

В процессе работы на дисках могут накапливаться файлы, содержащие уже ненужную информацию. Это могут быть различные временные, резервные или иные файлы. Постепенно накапливаясь, эти файлы могут бесполезно занимать большие объемы дисковой памяти. Поэтому нужно периодически подвергать ревизии имеющиеся на диске файлы, чтобы удалять все файлы, ставшие ненужными.

Для того чтобы выполнить очистку диска, нужно:

- 1) открыть окно *Свойства* выбранного диска;
- 2) перейти на страницу *Общие*;
- 3) нажать кнопку *Очистка диска*;
- 4) в окне *Очистка диска* просмотреть результаты (см. рис. 4.7, г).

В верхней части окна указано, какой именно объем дисковой памяти может быть освобожден в результате выполнения очистки.

С помощью кнопки *Просмотр файлов* можно просмотреть список файлов, которые подготовлены к удалению в группе. Итак, для фактического выполнения очистки в окне *Очистка диска* нужно:

- 1) выполнить анализ подготовленных к удалению файлов и групп файлов;
- 2) включить флажки тех групп файлов, которые следует удалить;
- 3) нажать кнопку *ОК* или клавишу [Enter];
- 4) после завершения очистки закрыть окно *Свойства очищенного диска*.

Контрольные вопросы

1. Какие виды форматирования вы знаете?
2. Как происходит процесс форматирования диска?
3. Перечислите способы копирования дискет.
4. Какие операции с дисками позволяет выполнять пункт *Свойства* контекстного меню диска?
5. Как можно увеличить объем свободного пространства на диске?
6. Какая информация содержится в окне *Результаты проверки диска*?

7. Как происходит процесс дефрагментации диска?
8. В процессе длительной эксплуатации на жестком диске могут возникнуть физические или логические дефекты. Какие из них могут быть исправлены?
9. Какая операция положительно сказывается на изнашиваемости механизма дисководов?
10. Увеличивает ли дефрагментация объем свободного пространства на диске?

4.3. ПРОГРАММЫ-АРХИВАТОРЫ

Одним из наиболее широко распространенных видов сервисных программ являются программы, предназначенные для архивации, упаковки файлов путем сжатия хранимой в них информации.

Сжатие информации — это процесс преобразования информации, хранящейся в файле, к виду, при котором уменьшается избыточность в ее представлении и, соответственно, требуется меньший объем памяти для хранения. Сжатие информации в файлах производится различными способами, например за счет упрощения кодов, исключения из них постоянных битов или представления повторяющихся символов или повторяющейся последовательности символов в виде коэффициента повторения. Применяются различные алгоритмы сжатия текстовой, графической и видеoinформации.

Сжиматься могут как один, так и несколько файлов, которые в сжатом виде помещаются в так называемый архивный файл, или архив.

Архивный файл — это специальным образом организованный файл, содержащий в себе один или несколько файлов в сжатом или несжатом виде, служебную информацию об именах файлов, дате и времени их создания или модификации, размерах и т.п.

Архивация — упаковка (сжатие) папки, файла или группы файлов — используется для того, чтобы уменьшить место, занимаемое ими на диске, при обмене информацией по сети Интернет, а также при создании резервных копий, например на дискетах. Для архивации используются специальные программы-архиваторы или диспетчеры архивов. Наибольшее распространение имеют способы сжатия данных в форматах *arj*, *zip* и *rar*.

Для операционной системы MS DOS чаще всего используют программы *Arj.exe*, *PKZip.exe/PKUNZip.exe* или *Rar.exe/UNRAR.exe*, а для Windows 98/NT/XP — *WinArj*, *WinZip*, *WinRAR*. Эти программы обеспечивают возможность использования и других архиваторов, поэтому если на компьютере, куда перенесены сжатые в них файлы, отсутствуют указанные программы, архивы можно распаковать с помощью другого архиватора.

Программы-архиваторы позволяют создавать и такие архивы, для извлечения из которых содержащихся в них файлов не требуются какие-либо программы, так как сами архивные файлы могут содержать программу распаковки. Такие архивные файлы называются самораспаковывающимися.

Самораспаковывающийся архивный файл — это загрузочный исполняемый модуль, который способен к самостоятельной разархивации находящихся в нем файлов без использования программы-архиватора.

Самораспаковывающийся архив получил название SFX-архив (Self-eXtracting). Архивы такого типа обычно создаются в форме *.exe-файла.

Если файл имеет большой объем, превышающий размер одной дискеты, то он может быть размещен на нескольких дискетах (томах). Для этого во многих программах-архиваторах предусмотрена возможность создания многотомных (распределенных) архивов. Каждый том в таком архиве является самостоятельной частью и ему присваивается порядковый номер.

Основными характеристиками программ-архиваторов являются:

- скорость работы;
- сервис (набор функций архиватора);
- степень сжатия — отношение размера упакованного файла к размеру исходного файла.

Степень сжатия файла характеризуется коэффициентом K_c , отражающим процентное отношение объема сжатого файла V_c к объему исходного файла V_0 :

$$K_c = \frac{V_c}{V_0} 100.$$

Степень сжатия зависит от используемой программы, метода сжатия и типа исходного файла. Наиболее хорошо сжимаются файлы графических образов, текстовые файлы и файлы данных, для которых степень сжатия может достигать 5...40 %; хуже сжимаются файлы исполняемых программ и загрузочных модулей — 60...90 %; почти не сжимаются архивные файлы.

Лучшие архиваторы позволяют реализовать следующие функции:

- создавать архивные файлы из отдельных (или всех) файлов текущего каталога и его подкаталогов, загружая в один архив до 32 000 файлов;
- добавлять файлы в архив;
- извлекать и удалять файлы из архива;
- просматривать содержимое архива;
- просматривать содержимое архивированных файлов;

- осуществлять поиск строк в архивированных файлах;
- вводить в архив комментарии к файлам, добавлять информацию о создателе архива, времени и дате последних изменений, внесенных в архив;
- создавать многотомные архивы;
- создавать самораспаковывающиеся архивы как в одном томе, так и в нескольких томах;
- запоминать в архиве пути к файлам;
- сохранять в архиве несколько поколений (версий) одного и того же файла;
- пересортировать архивный файл по размерам файлов, именам, расширениям, дате и времени модификации, коэффициенту сжатия и т.д.;
- обеспечивать защиту информации в архиве и доступ к файлам, помещенным в архив, защищать каждый из помещенных в архив файлов циклическим кодом;
- тестировать архив, проверять сохранность в нем информации, в том числе и с помощью паролей;
- восстанавливать файлы (частично или полностью) из разрушенных (поврежденных) архивов;
- поддерживать типы архивов, созданных другими архиваторами. Например, архиватор оболочки FAR Manager 1.6 поддерживает 14 архивных форматов.

Рассмотрим основные операции архивации файлов на примере WinRAR.

Программа WinRAR не входит в комплект поставки Windows. Для использования этого архиватора его необходимо предварительно установить на компьютер. При установке программы WinRAR происходит интеграция программы-архиватора с операционной системой, в результате чего в контекстных меню объектов появляются дополнительные команды, рядом с которыми находятся характерные значки программы WinRAR.

На рис. 4.8, а приведено контекстное меню, которое содержит команды *Добавить в архив...*, *Добавить в архив «picture.rar»*, *Добавить в архив и отправить по e-mail...*, *Добавить в архив «picture.rar» отправить по e-mail*.

В контекстном меню архивного файла *picture.rar* (рис. 4.8, б) появились команды *Извлечь файлы...*, *Извлечь в текущую папку*, *Извлечь в picture*.

Создать новый архив или добавить файлы в уже существующий архив с помощью программы WinRAR можно двумя способами.

Первый способ — создание архива с использованием контекстного меню (без предварительного запуска архиватора). Для его реализации необходимо выполнить следующие действия:

- 1) выделить файл (файлы) или папку для архивирования;

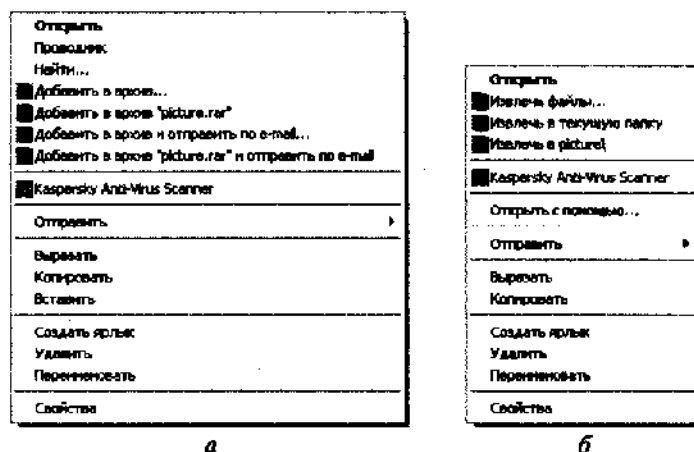


Рис. 4.8. Команды контекстных меню:

а — добавление файла в архив; *б* — извлечение файла из архива

2) вызвать контекстное меню и выбрать команду *Добавить в архив*, после чего откроется диалоговое окно *Имя и параметры архива*;

3) в диалоговом окне *Имя и параметры архива* ввести путь и имя архива (см. рис. 4.8).

4) щелкнуть по кнопке *OK*.

Второй способ — создание архива с предварительным запуском архиватора. Для его реализации необходимо выполнить следующие действия:

1) запустить программу WinRAR;

2) выделить файлы для архивирования;

3) ввести команду меню *Команды/Добавить в архив* или щелкнуть по кнопке *Добавить* на панели инструментов;

4) в диалоговом окне *Имя и параметры архива* ввести путь и имя архива (рис. 4.9);

5) щелкнуть по кнопке *OK*.

Используя диалоговое окно *Имя и параметры архива*, можно создавать самораспаковывающийся (SFX) или многотомный архив.

При открытии архива в окне WinRAR появится следующая информация (рис. 4.10):

- имена файлов, помещенных в архив;
- дата и время их создания;
- исходные размеры файлов;
- размеры файлов в сжатом виде;
- суммарный объем всех файлов, помещенных в архив;
- тип этих файлов.

Разархивировать файлы (извлечь из архива) можно тоже двумя способами.

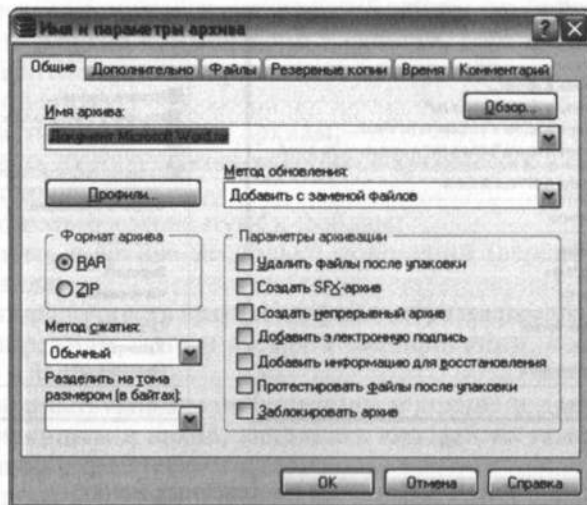


Рис. 4.9. Диалоговое окно *Имя и параметры архива*

Первый способ — без предварительного запуска архиватора:

- 1) открыть окно необходимой папки и выделить файл архива для распаковки;
- 2) вызвать контекстное меню и выбрать команду *Извлечь файлы*. Откроется диалоговое окно *Путь и параметры извлечения*;
- 3) в поле *Путь для извлечения* указать диск и папку, куда должны быть помещены распакованные файлы;
- 4) щелкнуть по кнопке *OK*.

Второй способ — с предварительным запуском архиватора:

- 1) запустить WinRAR;

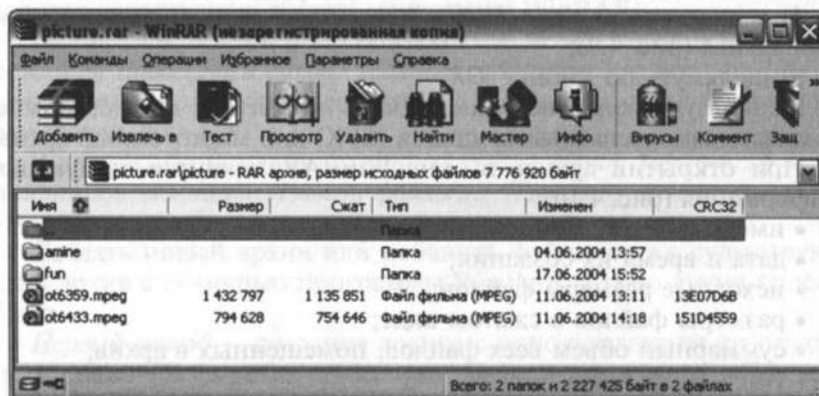


Рис. 4.10. Окно программы-архиватора WinRAR

2) ввести команду *Файл/Открыть архив* ([Ctrl] + [O]) либо открыть архив в дереве каталогов;

3) открыть папку, содержащую файл архива, и выбрать файл(ы), который(е) необходимо распаковать;

4) ввести команду *Команды/Извлечь в папку* либо щелкнуть по кнопке *Извлечь* на панели инструментов;

5) в открывшемся диалоговом окне *Путь и параметры извлечения* указать диск и папку, куда должны быть помещены распакованные файлы;

6) щелкнуть по кнопке *ОК*.

Программа WinRAR позволяет получать информацию о файлах, находящихся внутри архива, без распаковки архива, а также просматривать содержимое сжатых файлов без их извлечения из архива. Это важно, так как далеко не все архивы, поступающие на компьютер, особенно по сети Интернет, стоит распаковывать.

Для просмотра архивного файла необходимо выполнить следующие действия:

1) запустить WinRAR;

2) ввести команду *Файл/Открыть архив* ([Ctrl] + [O]) либо открыть архив в дереве каталогов;

3) сделать двойной щелчок по имени файла, который необходимо просмотреть. Файл откроется в окне программы, с которой он ассоциирован;

4) если нужно просмотреть файл в другой программе или неизвестно, с какой программой ассоциирован выделенный файл, то следует ввести команду *Команды/Просмотреть файлы* или щелкнуть по кнопке *Просмотр* на панели инструментов.

Контрольные вопросы

1. В чем заключается процесс архивирования (разархивирования) файлов?

2. Перечислите основные функции программ-архиваторов.

3. От чего зависит степень сжатия файлов?

4. Какие меры предусмотрены в программах-архиваторах для защиты информации и сохранения ее целостности?

5. Текстовый файл объемом 20 КБ помещен в архив и для него же создан самораспаковывающийся архив. Какой из вновь созданных файлов будет больше и почему?

4.4. КОМПЬЮТЕРНЫЕ ВИРУСЫ И АНТИВИРУСНЫЕ ПРОГРАММЫ

Первая массовая эпидемия компьютерного вируса произошла в 1986 г., когда вирус Brain заражал дискеты для первых персональных компьютеров. В настоящее время известно несколько де-

сятков тысяч вирусов, заражающих компьютеры с различными операционными системами и распространяющихся по компьютерным сетям.

Обязательным свойством компьютерного вируса является способность к размножению (самокопированию) и незаметному для пользователя внедрению в файлы, загрузочные секторы диска и документы. Название «вирус» взято из биологии именно по признаку способности к саморазмножению.

После заражения компьютера вирус может активизироваться и заставить компьютер выполнять какие-либо действия. Активизация вируса может быть связана с различными событиями (наступлением определенной даты или дня недели, запуском конкретной программы, открытием документа и т.д.).

Компьютерным вирусом называется специально написанная программа, способная самопроизвольно присоединяться к другим программам, создавать свои копии и внедрять их в файлы, системные области компьютера и вычислительные сети с целью нарушения работы программ, порчи файлов и каталогов, создания всевозможных помех в работе на компьютере.

Разнообразны последствия действия вирусов. По величине вредных воздействий вирусы можно подразделить:

- на неопасные, влияние которых ограничивается уменьшением свободной памяти на диске, графическими, звуковыми и другими внешними эффектами;
- опасные, которые могут привести к сбоям и зависаниям при работе компьютера;
- очень опасные, активизация которых может привести к потере программ и данных (изменению или удалению файлов и каталогов), форматированию винчестера и т.д.

При заражении компьютера вирусом очень важно своевременно его обнаружить. Для этого следует знать об основных признаках проявления вирусов. К ним можно отнести следующие:

- прекращение работы или неправильная работа ранее успешно функционировавших программ;
- медленная работа компьютера;
- невозможность загрузки операционной системы;
- исчезновение файлов и каталогов или искажение их содержимого;
- изменение даты и времени модификации файлов;
- изменение размеров файлов;
- неожиданное значительное увеличение числа файлов на диске;
- существенное уменьшение размера свободной оперативной памяти;
- вывод на экран не предусмотренных сообщений или изображений;

- подача не предусмотренных звуковых сигналов;
- частые зависания и сбои в работе компьютера.

По «среде обитания» вирусы можно подразделить на сетевые, файловые, загрузочные и макровирусы.

Сетевые вирусы. По компьютерной сети могут распространяться и заражать компьютеры любые обычные вирусы. Это может происходить, например, при получении зараженных файлов с серверов файловых архивов. Однако существуют и специфические сетевые вирусы, которые используют для своего распространения электронную почту и «всемирную паутину».

Интернет-черви (от *англ.* worm — червь) — это вирусы, которые распространяются в компьютерной сети во вложенных в почтовое сообщение файлах. Автоматическая активизация червя и заражение компьютера могут произойти при обычном просмотре сообщения. Опасность таких вирусов состоит в том, что они по определенным датам активизируются и уничтожают файлы на дисках зараженного компьютера.

Кроме того, интернет-черви часто являются троянами, выполняя роль «троянского коня», внедренного в операционную систему. Такие вирусы похищают идентификатор и пароль пользователя для доступа в Интернет и передают их на определенный почтовый адрес. В результате злоумышленники получают возможность доступа в Интернет за счет ничего не подозревающих пользователей.

Лавинообразная цепная реакция распространения вируса базируется на том, что вирус после заражения компьютера начинает рассылать себя по всем адресам электронной почты, которые имеются в адресной книге пользователя. Кроме того, может происходить заражение и по локальной сети, так как червь перебирает все локальные диски и сетевые диски с правом доступа и копируется туда под случайным именем.

Профилактическая защита от интернет-червей заключается в том, что не рекомендуется открывать вложенные в почтовые сообщения файлы, полученные из сомнительных источников.

Особой разновидностью вирусов являются активные элементы (программы) на языках JavaScript или VBScript, которые могут выполнять разрушительные действия, т.е. являться вирусами (скрипт-вирусами). Такие программы передаются по «всемирной паутине» в процессе загрузки Web-страниц с серверов Интернета в браузер локального компьютера.

Профилактическая защита от скрипт-вирусов состоит в том, что в браузере можно запретить получение активных элементов на локальный компьютер.

Файловые вирусы. Файловые вирусы различными способами внедряются в исполняемые файлы (программы) и обычно активизируются при их запуске. После запуска зараженной программы

вирус находится в оперативной памяти компьютера и является активным (т.е. может заражать другие файлы) вплоть до момента выключения компьютера или перезагрузки операционной системы. При этом файловые вирусы не могут заразить файлы данных (например, файлы, содержащие изображение или звук).

Профилактическая защита от файловых вирусов заключается в том, что не рекомендуется запускать на выполнение файлы, полученные из сомнительного источника и предварительно не проверенные антивирусными программами.

Загрузочные вирусы. Загрузочные вирусы записывают себя в загрузочный сектор диска. При загрузке операционной системы с зараженного диска вирусы внедряются в оперативную память компьютера. В дальнейшем загрузочный вирус ведет себя так же, как файловый, т.е. может заражать файлы при обращении к ним компьютера.

Профилактическая защита от таких вирусов заключается в отказе от загрузки операционной системы с гибких дисков и установке в BIOS вашего компьютера защиты загрузочного сектора от изменений.

Макровирусы. Макровирусы заражают файлы документов Word и электронных таблиц Excel. Макровирусы являются фактически макрокомандами (макросами), которые встраиваются в документ. После загрузки зараженного документа в приложение макровирусы постоянно присутствуют в памяти компьютера и могут заражать другие документы. Угроза заражения прекращается только после закрытия приложения.

Профилактическая защита от макровирусов заключается в предотвращении запуска вируса. При открытии документа в приложениях Word и Excel сообщается о присутствии в них макросов (потенциальных вирусов) и предлагается запретить их загрузку. Выбор запрета на загрузку макросов надежно защитит ваш компьютер от заражения макровирусами, однако отключит и полезные макросы, содержащиеся в документе. Для обнаружения, удаления и защиты от компьютерных вирусов разработано несколько видов специальных программ, которые позволяют обнаруживать и уничтожать вирусы. Такие программы называются антивирусными.

Различают следующие виды антивирусных программ:

- программы-детекторы;
- программы-доктора, или фаги;
- программы-ревизоры;
- программы-фильтры.

Программы-детекторы осуществляют поиск характерной для конкретного вируса последовательности байтов (сигнатуры вируса) в оперативной памяти и файлах и при обнаружении выдают соответствующее сообщение. Недостатком таких антиви-

русных программ является то, что они могут находить только те вирусы, которые известны разработчикам таких программ.

Программы-доктора, или фаги, а также программы-вакцины не только находят зараженные вирусами файлы, но и лечат их, т.е. удаляют из файла тело программы вируса, возвращая файлы в исходное состояние. В начале своей работы фаги ищут вирусы в оперативной памяти, уничтожая их, и только затем переходят к лечению файлов. Среди фагов выделяют полифаги, т.е. программы-доктора, предназначенные для поиска и уничтожения большого количества вирусов. Наиболее известными полифагами являются программы Adistest, Scan, Norton AntiVirus и Doctor Web.

Программы-ревизоры относятся к самым надежным средствам защиты от вирусов. Ревизоры запоминают исходное состояние программ, каталогов и системных областей диска тогда, когда компьютер не заражен вирусом, а затем периодически или по желанию пользователя сравнивают текущее состояние с исходным. Обнаруженные изменения выводятся на экран видеомонитора. Как правило, сравнение состояний производят сразу после загрузки операционной системы. При сравнении проверяются длина файла, код циклического контроля (контрольная сумма файла), дата и время модификации, другие параметры. Программы-ревизоры имеют достаточно развитые алгоритмы, обнаруживают стелс-вирусы и могут даже отличить изменения версии проверяемой программы от изменений, внесенных вирусом. К числу программ-ревизоров относится широко распространенная в России программа ADint фирмы «Диалог-Наука».

Программы-фильтры представляют собой небольшие резидентные программы, предназначенные для обнаружения подозрительных действий при работе компьютера, характерных для вирусов. Такими действиями могут являться:

- попытки коррекции файлов с расширениями COM и EXE;
- изменение атрибутов файлов;
- прямая запись на диск по абсолютному адресу;
- запись в загрузочные сектора диска;
- загрузка резидентной программы.

При попытке какой-либо программы произвести указанные действия фильтр посылает пользователю сообщение и предлагает запретить или разрешить соответствующее действие.

Одним из основных методов борьбы с вирусами является своевременная профилактика их появления и распространения. Признанным лидером в России в антивирусных технологиях является «Лаборатория Касперского».

Многие функциональные особенности практических современных антивирусов были впервые разработаны именно в этой компании. Ряд крупных западных производителей антивирусного программного обеспечения использует в своих продуктах антивирус-

ное ядро Антивируса Касперского. Исключительные надежность и качество программных продуктов «Лаборатории Касперского» подтверждаются многочисленными наградами, сертификатами российских и зарубежных компьютерных изданий, независимых тестовых лабораторий. Разработка и распространение основного продукта «Лаборатории Касперского» — Антивируса Касперского — началась в 1989 г.

Антивирусные решения «Лаборатории Касперского» AVP обеспечивают контроль над всеми потенциальными источниками проникновения компьютерных вирусов; они используются на рабочих станциях, файловых серверах, веб-серверах, почтовых шлюзах и межсетевых экранах, а также в карманных компьютерах. Удобные средства управления дают пользователям возможность максимально автоматизировать антивирусную защиту компьютеров и корпоративных сетей. В состав программных продуктов, разработанных компанией, входит персональный межсетевой экран, а также средства фильтрации нежелательных почтовых сообщений.

Наилучшей защитой компьютера, особенно если он не подключен к сети, является аккуратность и внимательность пользователя и соблюдение им следующих требований.

- Нужно использовать только лицензионные дистрибутивные копии программных продуктов, приобретать их следует только у официальных продавцов.

- Необходимо периодически проверять компьютер на наличие вирусов. Компьютер должен быть оснащен эффективным регулярно используемым и постоянно обновляемым пакетом антивирусных программ. Для обеспечения большей безопасности следует применять параллельно несколько антивирусных программ, например Adistest и Doctor Web, а также ежедневно использовать ревизор диска Adinf или программу AVP.

- Перед считыванием с дискет и компакт-дисков информации, записанной на других компьютерах, необходимо всегда проверять их на наличие вирусов. При переносе на компьютер файлов в архивированном виде после распаковки их тоже необходимо проверять.

- При работе на других компьютерах всегда нужно защищать свои дискеты от записи в тех случаях, когда на них не планируется запись информации.

- Необходимо периодически проверять жесткие диски компьютера, запуская антивирусные программы с защищенной от записи дискеты, предварительно загрузив операционную систему также с защищенной от записи системной дискеты.

- При работе в сетях необходимо использовать антивирусные программы для входного контроля всех файлов, получаемых из компьютерных сетей. Никогда не следует запускать непроверенные файлы, полученные по компьютерным сетям.

- Необходимо делать регулярное резервное копирование и периодически сохранять файлы, с которыми ведется работа, на внешнем носителе, например на дискетах. Архивные копии наиболее ценной информации лучше дублировать на разных дискетах.
- Нельзя оставлять дискеты в кармане дисководов при включении и перезагрузке операционной системы, чтобы исключить заражение компьютера загрузочными вирусами.

Контрольные вопросы

1. К каким последствиям может привести заражение компьютерными вирусами?
2. Какие типы компьютерных вирусов существуют, чем они отличаются друг от друга и какова должна быть профилактика заражения?
3. Сформулируйте правила обращения с дискетами.
4. Опишите основные виды антивирусных программ.
5. Как защитить компьютер от интернет-вирусов?
6. Почему даже чистая отформатированная дискета может стать источником заражения вирусом?
7. Составьте список известных вам вирусов, начиная с самых опасных.

3. Существует возможность обмена информацией между приложениями с помощью буфера обмена. Записанный в буфер фрагмент можно вставить либо в другое место того же документа, либо в другой документ того же приложения, либо в документ другого приложения (см. подразд. 4.1). Фрагмент, находящийся в буфере обмена, может вставляться произвольное число раз.

4. Общая технология связывания и внедрения объектов (технология OLE), которая позволяет:

- внедрять в документ объекты или фрагменты документов, созданные в других приложениях, а также редактировать эти объекты средствами создавшего их приложения;
- устанавливать связь объекта с документом другого приложения. При этом объект продолжает «жить собственной жизнью» и обслуживать другие документы.

5. Одинаково осуществляется завершение работы приложений. Перед завершением работы необходимо сохранить все документы и закончить работу одним из следующих способов:

- при помощи кнопки ;
- при помощи системного меню приложения;
- при помощи меню *Файл/Закреть*.

Контрольные вопросы

1. Что понимается под термином «информационные технологии»?
2. Перечислите операции по запуску приложений и открытию документов.
3. Как вы понимаете технологию связывания и внедрения объектов?
4. Люди каких профессий наиболее тесно связаны с информационными технологиями?

5.2. ТЕХНОЛОГИЯ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

Создавать, редактировать, форматировать и распечатывать текстовые документы можно с помощью специальных программ обработки текстов: текстовых редакторов и текстовых процессоров.

Текстовые редакторы — это программы для создания, редактирования, форматирования, сохранения и печати документов.

Текстовые процессоры — это текстовые редакторы, имеющие в своем составе расширенный набор функций для работы с документами.

Некоторые текстовые процессоры являются так называемыми WYSIWYG-редакторами (от *англ.* What You See Is What You Get — то, что ты видишь, есть то, что ты получишь). Когда говорят, что это WYSIWYG-редактор, то гарантируют полное соответствие внешнего вида документа на экране компьютера его печатной

копии. К редакторам такого типа относятся Word и StarWriter. Некоторые современные редакторы поддерживают концепцию «почти» WYSIWYG. Вид документа на экране при этом немного отличается от его печатной копии, но делается это специально с целью более эффективного использования рабочего окна документа. Примерами «почти» WYSIWYG-редакторов являются Netscape Composer и KLyX. Мощные программы обработки текста — настольные издательские системы — предназначены для подготовки документов к публикации. Примером такой системы является Adobe PageMaker.

Текстовые файлы — наиболее распространенный тип данных в компьютерном мире. Существуют несколько проблем, связанных с текстовыми файлами. Первая проблема — чрезвычайно большое число символов, требующихся для поддержки различных языков (см. подразд. 2.1). Вторая проблема заключается в том, что люди хотят, чтобы распечатываемые документы содержали графики, диаграммы, примечания, заголовки и чтобы при этом использовались различные шрифты. Кроме того, документы, распространяемые в Интернет (он-лайновые документы), содержат мультимедиа, ссылки на различные сетевые ресурсы и звуковое сопровождение.

Многие текстовые файлы передаются в виде простого текста. Простой текст сложно сделать привлекательным и легко читаемым, так как в нем нет шрифтов различных начертаний, графики, заголовков, подзаголовков и т.д. Эти дополнительные особенности называются разметкой текста. Говоря о разметке текста, выделяют понятия физической и логической разметки. При использовании физической разметки текста указывается точный вид каждого фрагмента: положение на странице, размер и начертание шрифта. При логической разметке указывается логическое значение данного фрагмента, например: «это заголовок главы». Эти два способа разметки предназначаются, как правило, для использования в разных ситуациях. Для того чтобы распечатать текст на принтере, необходимо использовать физическую разметку. Должны быть приняты решения о размере полей и абзацных отступах. Ранние версии текстовых процессоров использовали только физический тип разметки. При этом для каждого фрагмента указывается шрифт, размер и стиль.

При обмене информацией физическое оформление текста накладывает ряд ограничений, особенно для он-лайновых документов. Размер экрана, разрешение и шрифты различны для различных систем. По этим причинам все чаще используется логическое оформление текста. В некоторых случаях логическое оформление текста практически необходимо (при создании электронных документов типа страниц WWW или при создании и публикации объемных трудов, таких как книги).

Таблица 5.1

Основные объекты окна приложения

Объект	Назначение
Строка заголовка	Содержит название программы и название файла, который в данный момент обрабатывается
Меню	Важнейшая часть окна-приложения. Из него выбираются необходимые команды
Панель инструментов	Простым щелчком мыши на кнопке того или иного инструмента вы можете задать команду для редактора Word
Линейки	Служат для изменения абзацного отступа, масштаба изображения страниц текста на экране, ширины колонок текста и размеров ячеек таблиц, а также для установки позиции табуляции в тексте
Полосы прокрутки	Применяются в тех случаях, когда весь текст не помещается на экране и требует сдвига вверх-вниз или влево-вправо
Строка состояния	Выводит информацию о текущем положении текстового курсора в документе, текущем времени и текущем режиме редактирования

Для сохранения разметки документов при передаче текстовой информации от машины к машине применяют разные способы. Текстовые процессоры и издательские системы используют специально разработанные форматы файлов, содержащие не только текст, но и информацию о том, как он должен быть оформлен. Основная проблема заключается в несовместимости таких форматов, хотя наиболее сложные программы обычно могут читать файлы в форматах программ-конкурентов. Примерами такого подхода являются текстовые процессоры Word и StarWriter.

При другом подходе непосредственно в текст документа вставляются специальные команды разметки. Даже если у вас нет программного обеспечения, поддерживающего такой формат, вы все же сумеете в нем разобраться. Существует немало способов подобного представления разметки текста, в том числе:

- HyperText Markup Language (HTML), использующийся в World Wide Web;
- TeX и LaTeX, пользующиеся популярностью у многих академических изданий, а также у математиков, физиков, химиков и даже музыкантов.

Файлы, созданные разными редакторами, зачастую имеют уникальные расширения, позволяющие, не заглядывая внутрь документа, догадаться о способах разметки текста. Так, файлы, со-

зданными редакторами подготовки простого текста, часто имеют расширение .txt, а подготовленные в текстовом процессоре Word — расширение .doc. Кроме того, Word поддерживает и другие форматы, например RTF (расширение .rtf). Документы, содержащие команды разметки языка HTML, имеют расширение .html или .htm.

В процессе вывода символа на экран компьютера производится обратный процесс — декодирование. Одному и тому же двоичному коду в различных кодировках поставлены в соответствие различные символы. В большинстве случаев пользователь не должен заботиться о перекодировках текстовых документов, так как это делают специальные программы — конвекторы, встроенные в приложения.

Текстовый редактор Microsoft Word является приложением Windows, и в нем реализован объектный подход. Из табл. 5.1 и рис. 5.1 видно, что часть объектов повторяет стандартное окно Windows, но появились новые объекты, такие как строка состояния, панель рисования, панель инструментов, панель форматирования и т.д.



Рис. 5.1. Окно приложения Word

Прежде всего выясним, с какими объектами мы будем работать. Как и в других приложениях Microsoft Office, можно выделить две основные группы объектов: документы и инструменты.

Все документы MS Word создаются на основе выбранного шаблона (образца).

Шаблон — служебный файл, который содержит всевозможную информацию о структуре и оформлении документов конкретного типа.

Использование шаблонов позволяет создавать документы таким образом, чтобы все элементы оформления органично сочетались между собой.

Шаблоны являются специализированными программами и сохраняются как файлы с расширением .dot. В стандартный комплект пакета MS Office включены шаблоны для наиболее распространенных типов документов. При запуске Word без указания имени файла процессор по умолчанию предлагает начать создание нового документа под именем «Документ 1» на основе стандартного шаблона «Обычный». На рис. 5.2 показано, из каких объектов состоит объект *Документ*, а в табл. 5.2 приведены примеры свойств некоторых объектов.

Поясним некоторые термины.

Кегль (размер шрифта) — вертикальный размер, измеряемый в пунктах (1 пункт равняется 0,376 мм). Для большинства документов используются 10—12-пунктовые шрифты, в то время как газетная полоса может иметь только 8-пунктовые шрифты. Размер шрифта более 14 пунктов обычно используется лишь для заголовков и выделений.

Гарнитура (тип шрифта) определяет особенности написания целого набора символов, включающего в себя заглавные и строчные буквы, цифры, знаки пунктуации и специальные символы. Каждый тип шрифта имеет несколько стилей начертания символов (например, полужирный, курсив, полужирный курсив, обычный). Кроме того, можно ввести подчеркивание символов и фрагментов текста.

С помощью *инструментов* мы производим различные операции над документами. Так, оперируя различными символами, можно ввести текст, придать ему необходимый вид, расположить на странице.

Выполняя операции с фигурным текстом, можно оформить красивый заголовок на титульном листе. Операции с рисунками позволяют украсить внешний вид вашего документа и сделать его содержание более понятным.

Символ — это минимальный элемент текста. Он обладает следующими свойствами:

- размер;
- начертание (обычное, жирное, курсивное, подчеркивание);

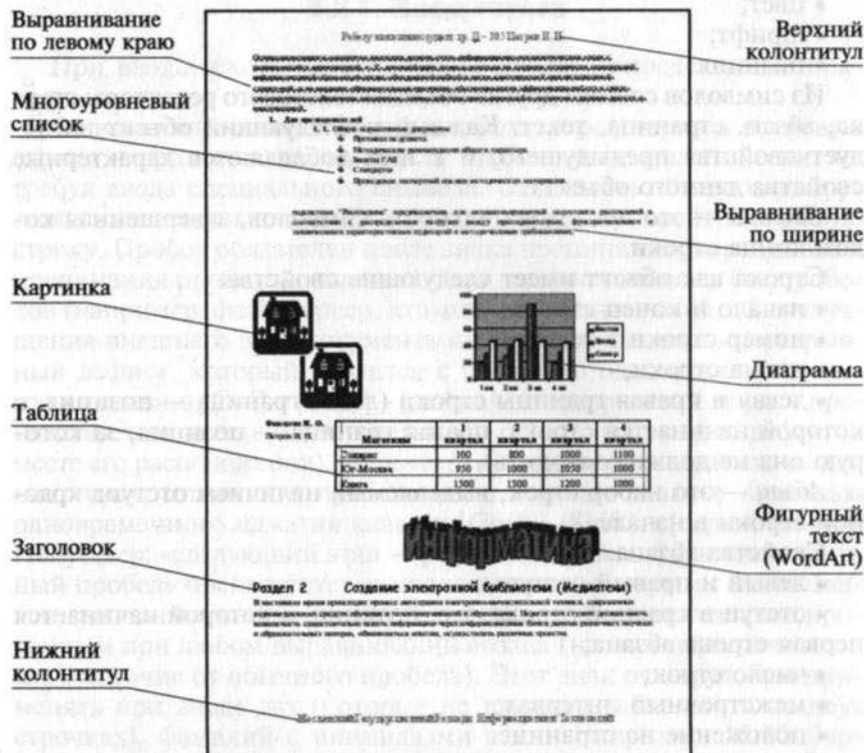


Рис. 5.2. Классификация объектов, составляющих объект *Документ*

Таблица 5.2

Свойства объектов

Объект	Свойства
Текст	1. Вид шрифта (кегель, гарнитура, цвет). 2. Стил абзаца. 3. Параметры страницы
Фигурный текст	1. Вид шрифта (кегель, гарнитура, цвет). 2. Формат текста. 3. Специальные эффекты (тень, наклон, поворот). 4. Размещение на странице
Рисунок	1. Формат графического файла. 2. Размер, масштаб. 3. Цветовое решение. 4. Размещение на странице
Таблица	1. Стил. 2. Число строк, столбцов. 3. Формат ячейки

- цвет;
- шрифт;
- позиция.

Из символов состоят другие объекты текстового редактора: строка, абзац, страница, текст. Каждый последующий объект наследует свойства предыдущего, и к ним добавляются характерные свойства данного объекта.

Строка — это последовательность символов, завершенная кодом конца строки.

Строка как объект имеет следующие свойства:

- начало и конец строки;
- номер строки в тексте;
- длина строки;
- левая и правая границы строки (левая граница — позиция, в которой начинается строка, правая граница — позиция, за которую она не должна заходить).

Абзац — это набор строк, выделяемый наличием отступа красной строки в начале.

Свойства абзаца:

- левый и правый отступы;
- отступ в красной строке, т.е. позиция, с которой начинается первая строка абзаца;
- число строк;
- межстрочный интервал;
- положение на странице.

Страница — это набор строк, завершаемый кодом конца страницы. Ее свойства:

- номер страницы;
- число строк на странице;

Последним, глобальным, объектом текстового редактора является весь текст в целом. Ко всем наследуемым свойствам у текста добавляются новые:

- начало и конец текста;
- число строк в тексте;
- расположение текста на странице.

Итак, для того чтобы подготовить документ для печати, необходимо научиться выполнять следующие операции:

- ввод текста;
- редактирование текста;
- форматирование символов;
- форматирование абзацев;
- вставка иллюстраций;
- вставка таблиц;
- вставка колонтитулов;
- сохранение документа;
- печать.

5.2.1. Ввод текста

При вводе текста необходимо соблюдать определенные правила.

Во всех современных текстовых редакторах переход на новую строку в процессе набора текста происходит автоматически, не требуя ввода специального символа. Окончание абзаца маркируется нажатием клавиши [Enter], позволяющей перейти на новую строку. Пробел обязателен после знака препинания; перед знаком препинания пробел не ставится. Знак «дефис» ставится без пробелов (например: файл-сервер, кто-нибудь, темно-синий). Для улучшения внешнего вида документа можно использовать «неразрывный дефис», который вводится с помощью одновременного нажатия клавиш [Ctrl] + [Shift] + [дефис]. «Неразрывный дефис» препятствует нежелательному переносу слов, содержащих дефис (в месте его расположения). Знак «тире», или как его называют «длинное тире», ставится с пробелами до и после знака с помощью одновременного нажатия клавиш [Ctrl] + [Shift] + [серый минус]. Например: «следующий этап — редактирование». Знак «неразрывный пробел» препятствует символам, между которыми он поставлен, располагаться на разных строчках и сохраняется фиксированным при любом выравнивании абзаца (не может увеличиваться, в отличие от обычного пробела). Этот знак очень удобно применять при вводе дат (которые не принято располагать на двух строчках), фамилий с инициалами и т.п. Например: М.Ю.Лермонтов, 1998 г., 9 мая, 1682—1696 гг. Ставится знак «неразрывный пробел» с помощью одновременного нажатия комбинации клавиш [Ctrl] + [Shift] + [пробел]. Слова, заключенные в кавычки или скобки, не должны отделяться от них пробелами. Например: «Евгений Онегин», гостиница «Невский Палас», слобода Алексеевка (ныне г. Алексеевка Белгородской обл.). Для ввода римских цифр используются прописные латинские буквы I, V, X, L, C, D, M. Например: первая половина XIX в., конец XVII в.

5.2.2. Редактирование текста

Редактирование — это проверка и исправление текста при подготовке его к печати.

Способы работы с фрагментами текста. Фрагментом называется непрерывная часть текста. Выделение фрагмента делает его объектом последующей команды.

Выделить фрагмент — значит обозначить его инверсным цветом («подсветить») с помощью мыши или клавиатуры.

Выделение фрагмента текста. Выделять можно:

- весь текст: команда *Правка/Выделить все*;

- несколько последовательно расположенных фрагментов текста: [Shift] + клавиша перемещения курсора;
- несколько произвольно расположенных фрагментов текста: [Ctrl] + мышь для выделения фрагментов;
- объект (рисунок, формулу, диаграмму): установка курсора на объекте и одинарный щелчок левой клавишей мыши.

Копирование, перемещение и удаление фрагментов текста. Эти операции выполняются только по отношению к выделенному фрагменту текста. При этом можно использовать два независимых механизма: через буфер обмена (см. подразд. 4.1) и методом drag-and-drop (перетаскивание мышью). В Word работа с буфером обмена дублируется специальными кнопками панели инструментов *Стандартная*:

 — копировать в буфер;  — вырезать;  — вставить из буфера.

Для перемещения выделенного фрагмента надо на нем установить указатель мыши и, удерживая нажатой левую кнопку, переместить мышь. Для копирования выделенного фрагмента нажимается клавиша [Ctrl] и выполняется перемещение фрагмента на новое место.

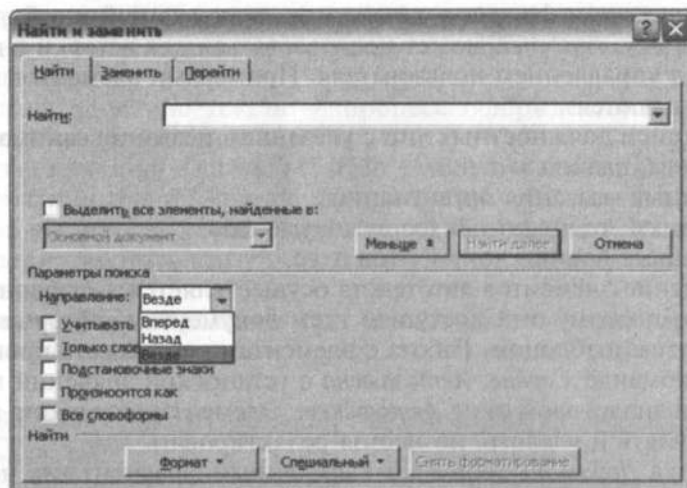
Поиск и замена фрагментов текста. В документах Word можно осуществлять поиск и замену текста, форматов (шрифта, абзаца, языка, стиля), различных специальных символов (маркера абзаца, сноски или примечания, конца раздела, колонки и т.п.). Для этого используют команды *Правка, Найти* (только поиск) или *Правка, Заменить* (поиск и замена).

На вкладке *Найти* окна *Найти и заменить* (рис. 5.3, а) указывается образец поиска. Объектом поиска является группа символов с учетом или без учета формата — шрифт, абзац, язык, стиль. Кроме того, можно задать направление поиска по отношению к текущей установке курсора в тексте документа в окне *Направление* (везде, вперед, назад).

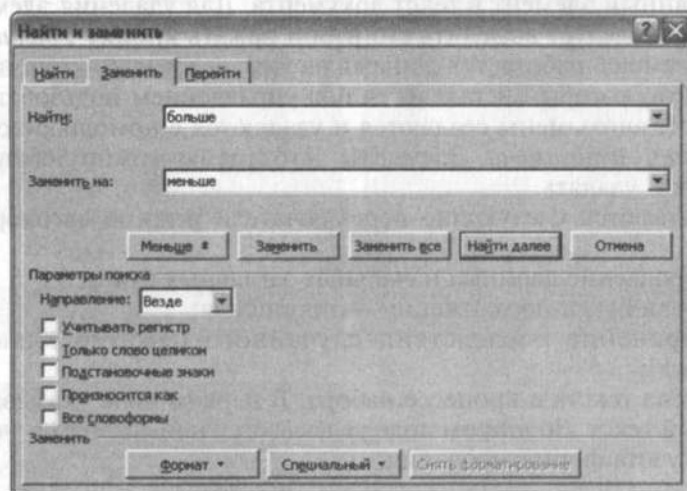
Кнопка *Найти далее* ищет следующее вхождение образца, указанного в поле *Найти*. Вкладка *Заменить* окна *Найти и заменить* (рис. 5.3, б) заменяет найденный образец на образец, помещенный в поле *Заменить на*, и ищет следующее вхождение. Кнопка *Заменить все* ищет все вхождения образца из поля *Найти* и заменяет его на образец из поля *Заменить на* без предварительных запросов.

Проверка орфографии. Для проверки орфографии необходимо выполнить команду горизонтального меню *Сервис/Правописание*.

Проверка орфографии выполняется по основному словарю выбранного языка и по дополнительным словарям пользователя, которые могут содержать произвольные слова (например, условные обозначения, ключевые слова языка программирования и др.).



а



б

Рис. 5.3. Поиск и замена фрагментов текста:

а — вкладка *Найти*; б — вкладка *Заменить*

Можно создать новый словарь пользователя либо отредактировать существующий словарь. Проверка орфографии охватывает основной текст, текст колонтитулов, сносок, концевых сносок и примечаний. Для задания параметров проверки текста можно воспользоваться кнопкой *Параметры* окна *Правописание*.

Приведем дополнительные возможности редактора Word по автоматической замене фрагментов текста.

Автотекст — фрагмент документа, включающий в себя текст или графику, который может использоваться для вставки в документ под управлением пользователя. Примерами элементов автотекста являются:

- подписи должностных лиц с указанием названия занимаемой должности, звания и т. п.;
- полные названия организаций;
- «шапки» стандартных форм документов;
- типовые бланки документов и др.

Хранение элементов автотекста осуществляется в шаблоне документа, поэтому они доступны всем документам, которые связаны с этим шаблоном. Работа с элементами автотекста происходит по команде *Сервис, Автозамена* с установкой значений параметров в диалоговом окне *Автотекст*. Элементы автотекста можно добавлять и удалять, но нельзя редактировать.

Кнопка *Добавить* добавляет выделенный фрагмент как новый элемент автотекста с заданным именем. Кнопка *Вставка* вставляет выбранный элемент в текст документа. Для удаления элемента автотекста следует выделить его имя и нажать кнопку *Удалить*.

Автозамена работает в динамическом режиме (в отличие от автотекста, который вставляется под управлением пользователя). Элементы автозамены создаются и удаляются с помощью команды *Сервис, Автозамена*. Элементы автозамены можно добавлять, заменять и удалять.

Указываются следующие переключатели режима автокоррекции:

- исправление двойных начальных заглавных букв;
- первая буква предложения — прописная;
- устранение последствий случайного нажатия клавиши [CapsLock];
- замена текста в процессе набора. В первом поле указывается исходный текст. Во втором поле указывается новый текст с учетом или без учета форматирования.

5.2.3. Форматирование текста

Читая газеты, журналы, книги, знакомясь с разнообразной печатной продукцией, мы видим различные способы расположения и оформления текста на странице. Один и тот же шрифт может иметь различные выделения: полужирный, курсив, подчеркнутый. Помимо различий в изображении шрифта существуют различия и в способах оформления абзацев. В печатных изданиях можно встретить абзацы, выровненные по левому краю или по обоим краям. Строки абзаца могут располагаться на разном расстоянии друг от друга (взаимное расположение строк в абзаце называется

«интерлиньяж»). Существует много возможностей оформления страниц с использованием различных способов оформления шрифта и абзацев.

Форматирование текста — процесс оформления страницы, абзаца, строки, символа.

В текстовом процессоре MS Word существуют два способа форматирования текста: прямое форматирование выделенных фрагментов и стилевое форматирование, с помощью которого можно переформатировать сразу несколько абзацев без предварительного их выделения.

Прямое форматирование. Под *форматированием шрифта* понимается изменение параметров введенных символов и абзацев. Word

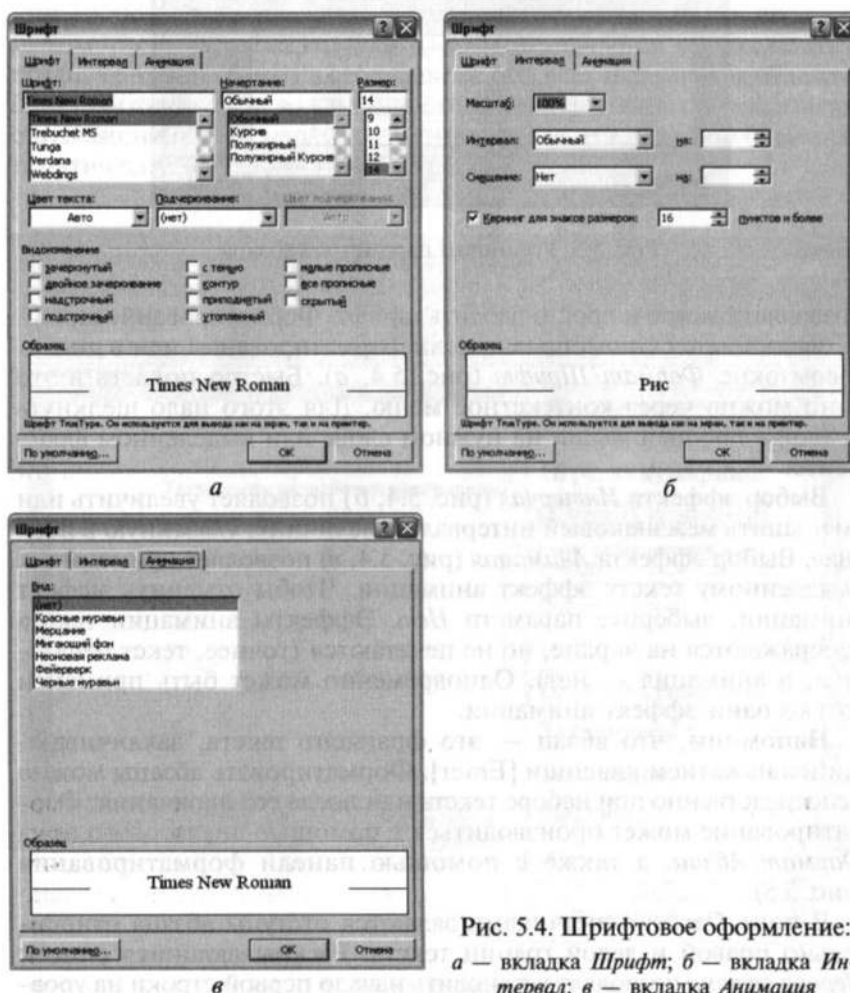


Рис. 5.4. Шрифтовое оформление:
а — вкладка *Шрифт*; б — вкладка *Интервал*; в — вкладка *Анимация*

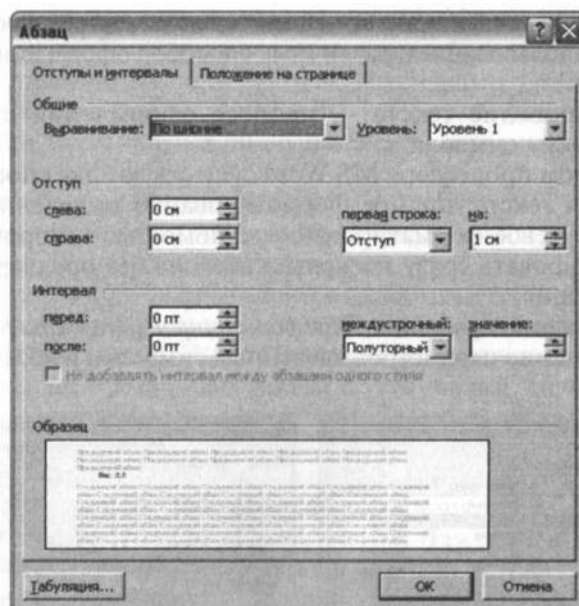


Рис. 5.5. Установка параметров абзаца

позволяет быстро и просто вводить шрифт. Форматирование шрифта выполняется с помощью панели форматирования или в диалоговом окне *Формат/Шрифт* (рис. 5.4, а). Быстро попасть в это окно можно через контекстное меню. Для этого надо щелкнуть правой клавишей мыши на нужном слове или выделенном фрагменте.

Выбор эффекта *Интервал* (рис. 5.4, б) позволяет увеличить или уменьшить межзнаковый интервал на величину, указанную в поле «на». Выбор эффекта *Анимация* (рис. 5.4, в) позволяет применить к выделенному тексту эффект анимации. Чтобы отменить эффект анимации, выберите параметр *Нет*. Эффекты анимации текста отображаются на экране, но не печатаются (точнее, текст печатается, а анимация — нет). Одновременно может быть применен только один эффект анимации.

Напомним, что абзац — это фрагмент текста, заканчивающийся нажатием клавиши [Enter]. Форматировать абзацы можно непосредственно при наборе текста или после его окончания. Форматирование может производиться с помощью диалогового окна *Формат/Абзац*, а также с помощью панели форматирования (рис. 5.5).

В поле *Отступ* счетчиками задаются отступы абзаца относительно правой и левой границ текста. Раскрывающийся список *Первая строка* позволяет установить начало первой строки на уров-

не левой границы абзаца, сформировать красную строку в виде отступа или выступа. В поле *Интервал* можно установить расстояние между форматировемым абзацем и соседними с ним. Для этого используются счетчики *Перед* и *После*. Раскрывающийся список *Межстрочный* (*Междустрочный*) позволяет изменить расстояние между строками абзаца (интерлиньяж). В раскрывающемся списке *Выравнивание* можно выбрать один из четырех вариантов: по левому или правому краю, по ширине (по обоим краям) или по центру.

Для абзацев можно изменять параметры шрифта, как было рассмотрено выше.

Стилевое форматирование. Стилиевое форматирование значительно упрощает форматирование документов. Смысл этой операции заключается в том, что абзацам определенного внешнего вида (в том числе заголовкам) или словам назначается определенный стиль, который хранит в себе данные обо всех параметрах объекта. При необходимости изменения стиля всех подобных объектов достаточно изменить параметры стиля, и все эти объекты будут автоматически переформатированы.

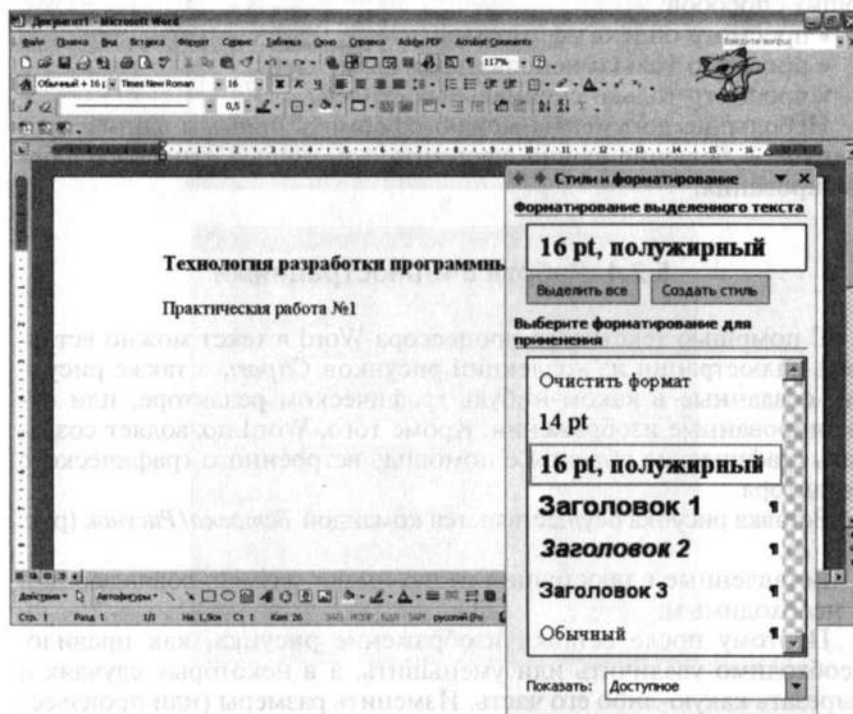


Рис. 5.6. Диалоговое окно *Стили*

Таким образом, отпадает необходимость изменять вручную формат каждого объекта, что существенно экономит время подготовки документа. Особенно удобно применять стилевое форматирование в больших по объему документах. Небольшие документы можно оформить прямым форматированием.

Стилевое форматирование — назначение специальных стилей символам или абзацам.

Стили хранятся вместе с документами или их шаблонами. Необходимый стиль можно выбрать либо из раскрывающегося списка *Стили/Панели форматирования*, либо с помощью команды *Формат/Стиль*.

По умолчанию этот список содержит имена только тех стилей, которые использовались в данном документе. Чтобы просмотреть список всех стилей, необходимо при раскрытии основного списка нажать клавишу [Shift].

Можно изменять и удалять существующие стили, можно создавать новые. Для совершения этих операций используется команда *Формат/Стиль*. Эта команда вызывает диалоговое окно *Стиль* (рис. 5.6).

В этом окне можно просмотреть список стилей одним из следующих способов:

- просмотр списка всех стилей;
- просмотр только использующихся стилей;
- просмотр только специальных стилей.

Небольшие документы можно оформить прямым форматированием, а большие лучше оформить с помощью стилового форматирования.

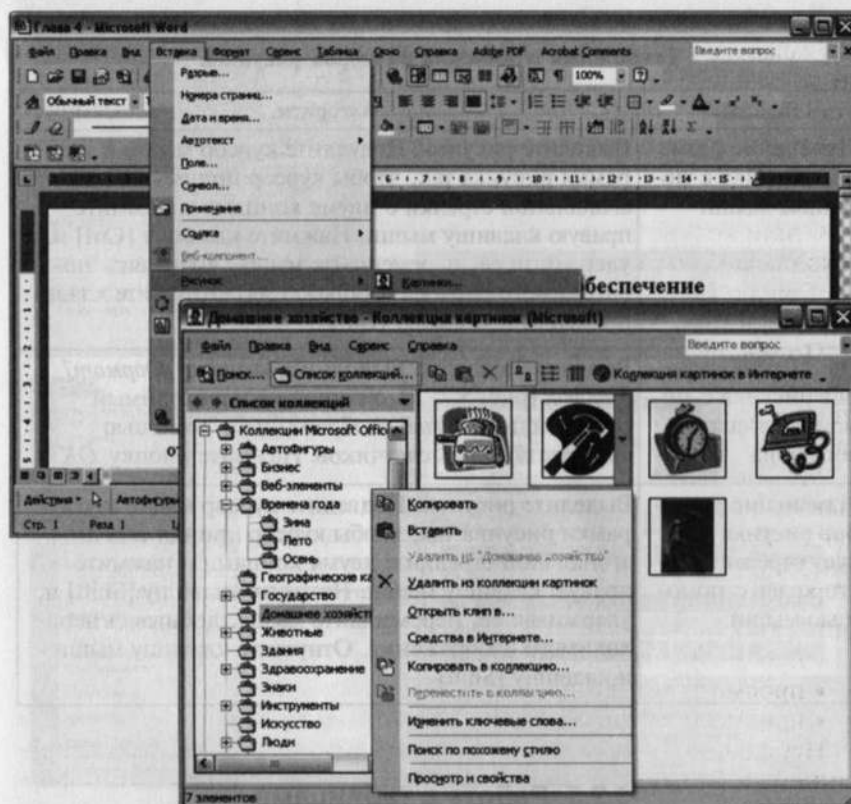
5.2.4. Работа с иллюстрациями

С помощью текстового процессора Word в текст можно вставлять иллюстрации из коллекций рисунков *Clipart*, а также рисунки, созданные в каком-нибудь графическом редакторе, или отсканированные изображения. Кроме того, Word позволяет создавать графические объекты с помощью встроенного графического редактора.

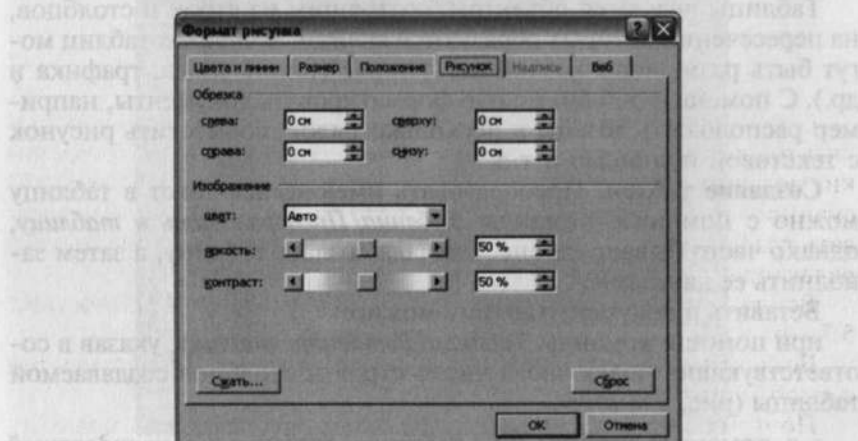
Вставка рисунка осуществляется командой *Вставка/Рисунок* (рис. 5.7, а).

Вставленные иллюстрации редко имеют размер, совпадающий с необходимым.

Поэтому после вставки изображение рисунка, как правило, необходимо увеличить или уменьшить, а в некоторых случаях и вырезать какую-либо его часть. Изменить размеры (или произвести масштабирование) рисунка можно, используя приведенные в табл. 5.3 алгоритмы.



а



б

Рис. 5.7. Работа с рисунками:

а — вставка из коллекции CLIPART; б — формат рисунка

Таблица 5.3

Технология изменения размеров рисунков

Действие	Алгоритм
Изменение размеров рисунка с помощью мыши	Выделите рисунок. Подведите курсор мыши к углу рамки рисунка так, чтобы курсор принял вид диагональной стрелки с двумя концами и нажмите правую клавишу мыши. Нажмите клавишу [Ctrl] и, удерживая ее, перемещайте мышь, добиваясь необходимого размера изображения. Отпустите клавишу мыши и клавишу [Ctrl]
Изменение размеров рисунка с помощью диалогового окна	Выделите рисунок. Выполните команду <i>Формат/Рисунок</i> (рис. 5.7, б). Установите необходимый размер или масштаб изображения с помощью соответствующих счетчиков. Нажмите кнопку <i>ОК</i>
Изменение размеров рисунка за счет отрезания его краев с помощью мыши	Выделите рисунок. Подведите курсор мыши к углу рамки рисунка так, чтобы курсор принял вид диагональной стрелки с двумя концами и нажмите правую клавишу мыши. Нажмите клавишу [Shift] и, удерживая ее, перемещайте мышь, добиваясь необходимого изображения. Отпустите клавишу мыши и клавишу [Shift]

5.2.5. Работа с таблицами

Таблицы являются объектом, состоящим из строк и столбцов, на пересечении которых образуются ячейки. В ячейках таблиц могут быть размещены различные данные (текст, числа, графика и др.). С помощью таблиц можно форматировать документы, например расположить абзацы в несколько рядов, совместить рисунок с текстовой подписью и т.д.

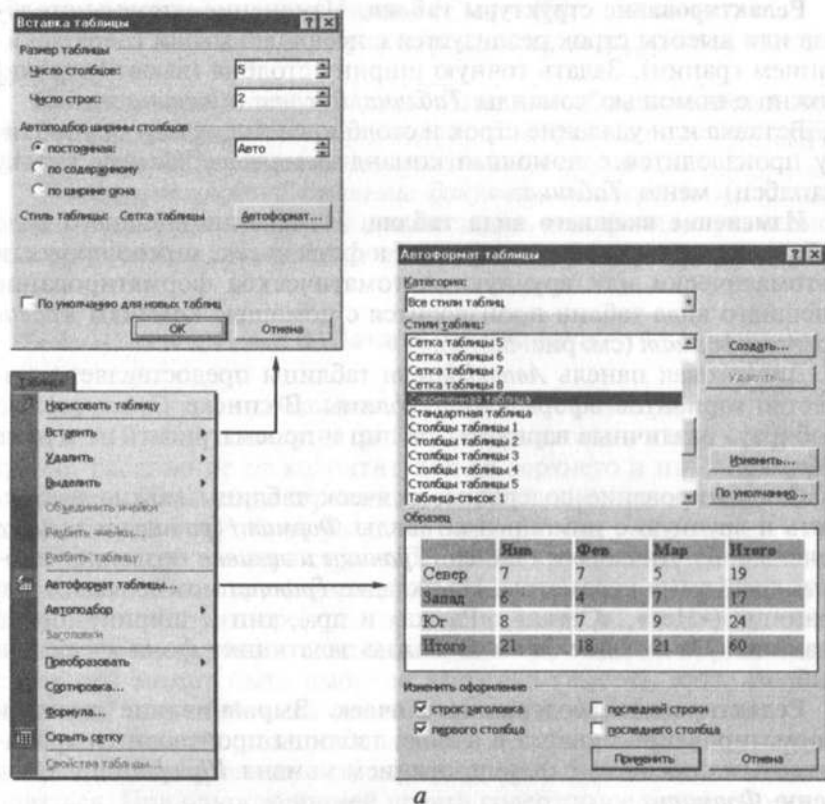
Создание таблиц. Преобразовать имеющийся текст в таблицу можно с помощью команды *Таблица/Преобразовать в таблицу*, однако часто бывает удобнее сначала создать таблицу, а затем заполнить ее данными.

Вставить в документ таблицу можно:

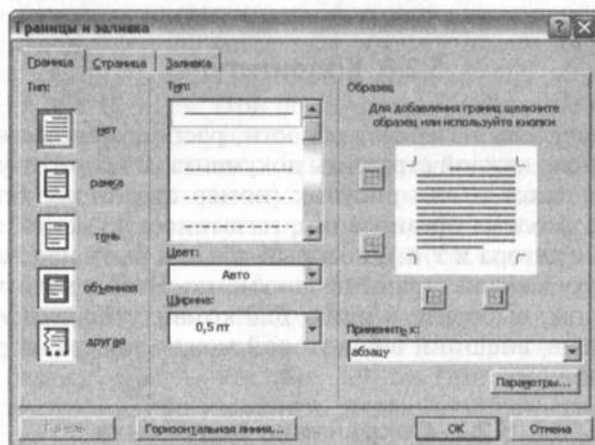
при помощи команды *Таблица/Вставить таблицу*, указав в соответствующих полях ввода число строк и столбцов создаваемой таблицы (рис. 5.8, а);

- с помощью кнопки  (*Добавить таблицу на стандартной панели*);

- а также можно нарисовать таблицу с помощью мыши при активизированной кнопке .



а



б

Рис. 5.8. Вставка и автоформат таблицы:

а — вставка таблицы; б — форматирование таблицы

Редактирование структуры таблиц. Изменение ширины столбцов или высоты строк реализуется с помощью мыши (перетаскиванием границ). Задать точную ширину столбца (высоту строки) можно с помощью команды *Таблица/Высота и ширина ячейки*.

Вставка или удаление строк и столбцов в именуемую таблицу производится с помощью команд *Вставить/Удалить строку* (столбец) меню *Таблица*.

Изменение внешнего вида таблиц. Изменение внешнего вида таблиц, т. е. установку вида границ и фона ячеек, можно провести автоматически или вручную. Автоматическое форматирование внешнего вида таблиц производится с помощью команды *Таблица/Автоформат* (см. рис. 5.8, а).

Диалоговая панель *Автоформат* таблицы предоставляет множество вариантов оформления таблицы. В списке *Стили* можно выбирать различные варианты таблиц и просматривать их в окне *Образец*.

Форматирование содержимого ячеек таблицы можно выполнить и вручную с помощью команды *Формат/Границы и заливка* (рис. 5.8, б). Диалоговая панель *Границы и заливка* позволяет выбрать требуемые параметры. На вкладке *Граница* можно задать тип границы («Нет», «Сетка», «Рамка» и др.), тип и ширину линий границы. На вкладке *Заливка* можно задать цвет фона ячеек или выбрать узор.

Редактирование содержимого ячеек. Выравнивание текста и форматирование шрифта в ячейке таблицы производится традиционным способом с использованием команд *Шрифт* или *Абзац* меню *Формат*.

5.2.6. Колонтитулы

Колонтитулами называют области, расположенные в верхнем и нижнем поле каждой страницы документа. В колонтитул обычно вставляется текст и/или рисунок (номер страницы, дата печати документа, эмблема организации, название документа, имя файла, фамилия автора и т. п.), который должен быть напечатан внизу иливерху каждой страницы документа. Чтобы начать работу с колонтитулом, выберите в меню *Вид* команду *Колонтитулы*; появится панель, внешний вид которой можно видеть на рис. 5.1.

5.2.7. Сохранение документа

Сохранить новый документ при работе с Word можно одним из следующих способов:

- выбрать команду меню *Файл/Сохранить как*;

- нажать комбинацию клавиш [Ctrl] + [S];
- нажать кнопку с изображением дискеты на панели инструментов.

При использовании указанных выше способов появляется диалоговое окно *Сохранение документа*, если файл документа сохраняется первый раз и вы еще не назначили ему имя.

Диалоговое окно *Сохранение документа* содержит несколько полей, с помощью которых можно задать имя файла, путь и диск для сохранения документа, а также форматы сохранения.

5.2.8. Печать документа

Перед печатью необходимо установить параметры страниц документа. К ним относятся размер и ориентация листа, размеры полей, расстояние от колонтитулов до верхнего и нижнего краев страницы. Параметры страниц могут быть одинаковыми для всего документа, а могут быть различными в разных разделах.

При выборе размера страницы ориентируйтесь на печатающее устройство. Так, лист формата A4 можно распечатать на любом из современных принтеров, тогда как формат A3 поддерживается только специальными модификациями принтеров. Ориентация страницы может быть выбрана книжная (высота листа больше ширины) или альбомная (высота листа меньше ширины).

При установке размеров полей прежде всего необходимо определить, односторонняя или двухсторонняя печать будет использоваться. При односторонней печати содержимое печатного издания располагается только на одной стороне печатного листа (такая печать часто используется при оформлении докладов, рефератов, диссертаций и т. п.). В этом случае различают верхнее, нижнее, левое и правое поля. При подготовке журнальных и книжных изданий используется двухсторонняя печать. Здесь различают так называемые зеркальные поля, а также вместо левого и правого полей используют понятия внутренних и внешних полей.

Для установки параметров страницы используют команду *Параметры страницы* меню *Файл*. При этом открывается диалоговое окно *Параметры страницы* (рис. 5.9). В этом окне три раздела: «Поля», «Размер бумаги», «Источник бумаги».

Размер и вид полей, а также положение переплета устанавливаются в разделе «Поля» (см. рис. 5.9, а). При установке размера полей необходимо либо увеличить размер внутреннего (или левого) поля, размер переплета, либо установить размер переплета отдельно. Размер и ориентация бумаги устанавливаются в разделе «Размер бумаги» (см. рис. 5.9, б). В разделе «Источник бумаги» устанавливается положение колонтитулов и выравнивание страницы (см. рис. 5.9, в).

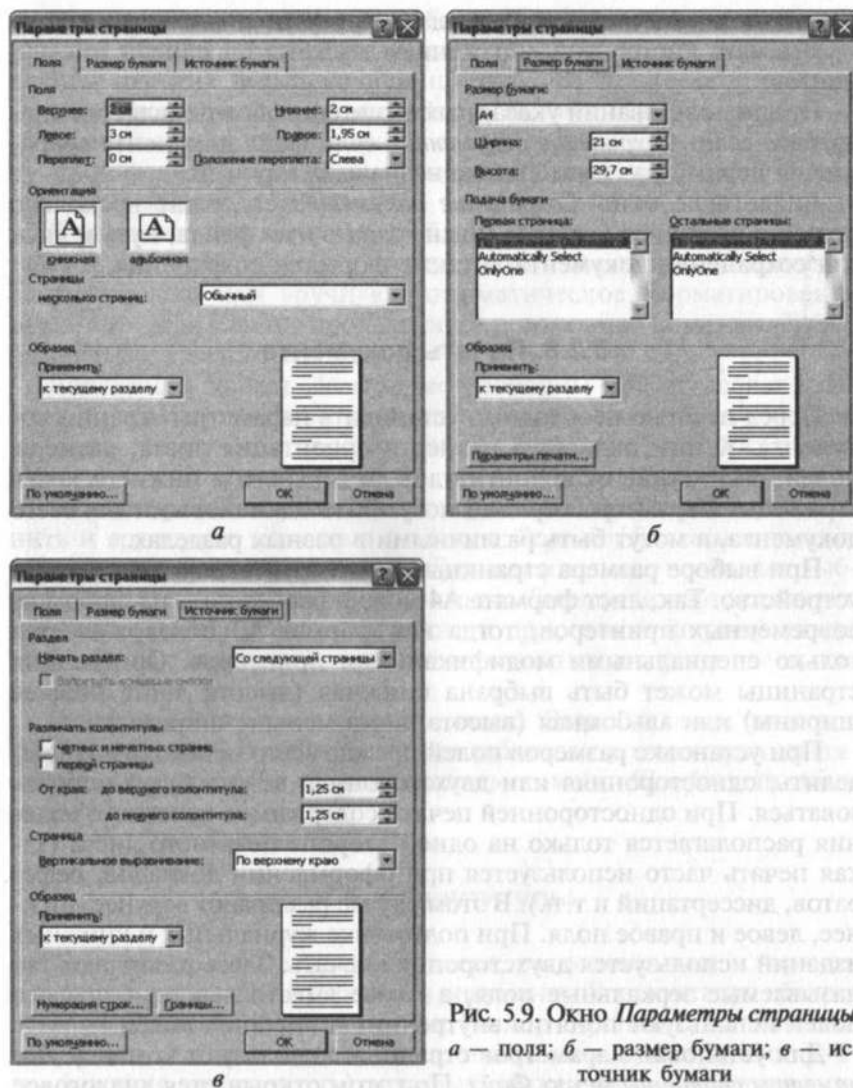
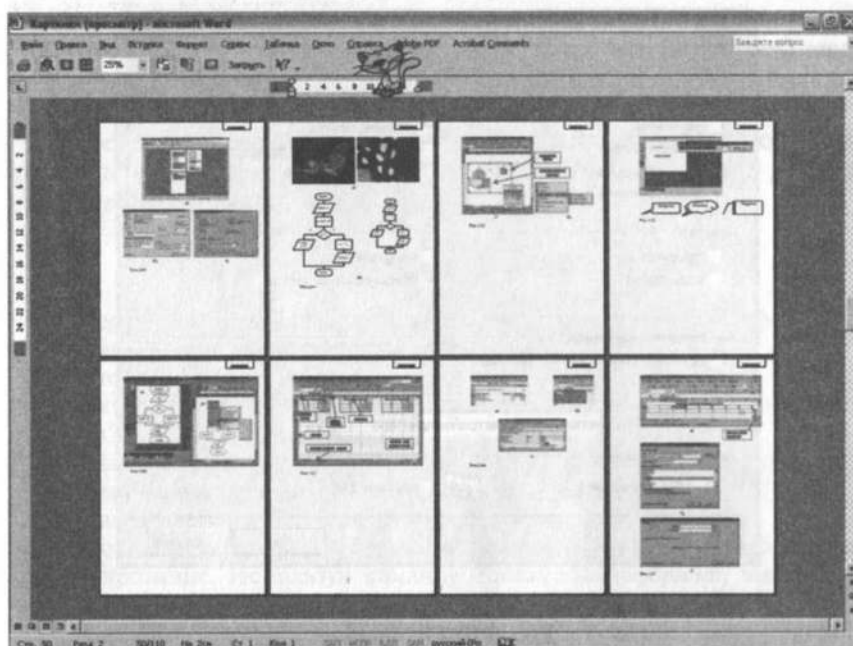


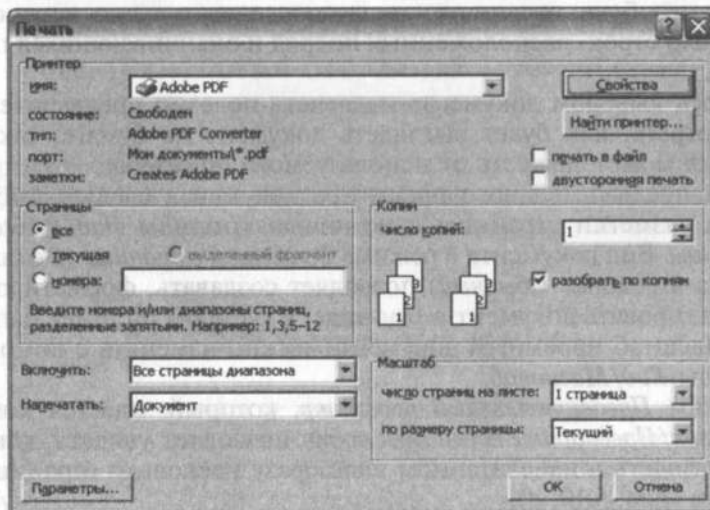
Рис. 5.9. Окно *Параметры страницы*:
а — поля; б — размер бумаги; в — источник бумаги

При выравнивании текста по ширине могут возникать слишком большие интервалы между словами. Чтобы этого избежать, необходимо установить переносы в словах. Если же в начале работы с документом задать функцию автопереноса, то этот процесс автоматизируется. Автоперенос задается в диалоговом окне *Расстановка переносов*, которое вызывается командой *Расстановка переносов* в меню *Сервис*.

Для задания функции автопереноса необходимо установить флажок «Автоматическая расстановка переносов». Диалоговое окно



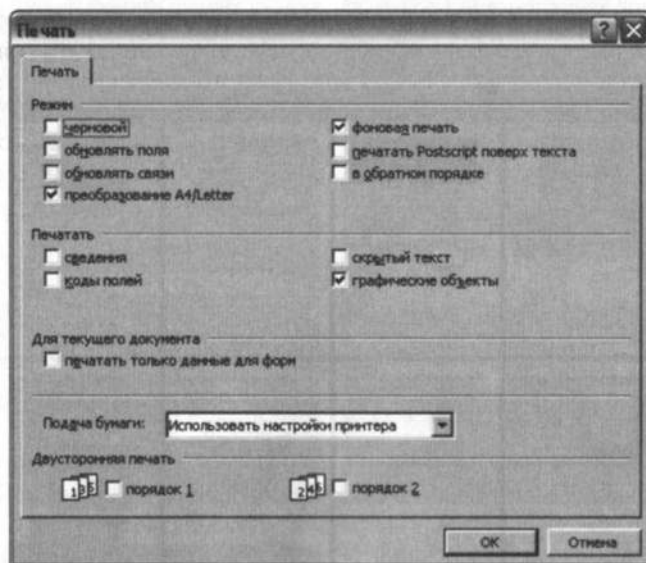
a



б

Рис. 5.10. Печать документа:

a — диалоговое окно *Предварительный просмотр*; б — диалоговое окно *Печать*;
в — диалоговое окно, открываемое кнопкой *Параметры* (см. также с. 202)



б

Рис. 5.10. Окончание

позволяет также разрешить или запретить переносы в словах из прописных букв, ограничить число последовательных переносов, т. е. число строк, расположенных подряд и оканчивающихся переносом.

Перед выводом документа на печать полезно предварительно просмотреть, как будет выглядеть документ на бумаге, так как этот вид может зависеть от используемого принтера.

Для предварительного просмотра документа следует выбрать режим разметки страницы с помощью команды *Вид/Разметка страницы*. Вид документа в режиме *Разметка страницы* (в отличие от вида в режиме *Обычный*) позволяет создавать, форматировать и редактировать документ в том виде, в котором он будет напечатан. Масштаб просмотра документа можно изменять с помощью команды *Вид/Масштаб*.

Режим *Предварительный просмотр*, который задается командой *Файл/Предварительный просмотр*, позволяет увидеть, как будет выглядеть в напечатанном виде сразу несколько страниц документа (рис. 5.10, а).

При подготовке документа к печати необходимо установить параметры печати, например номера страниц выводимых на печать, число копий и др. Команда *Файл/Печать* вызывает диалоговую панель *Печать*, которая позволяет выбрать принтер, число копий и номера страниц, выводимых на печать (рис. 5.10, б).

Кроме того, используя кнопку *Параметры*, можно задать дополнительные функции для печати, например печатать графические объекты или осуществлять двухстороннюю печать — сначала все нечетные страницы, а затем четные (рис. 5.10, в).

Перед выводом документа на печать необходимо настроить параметры печати и обязательно проверить их настройку в режиме *Предварительный просмотр*.

Контрольные вопросы

1. Перечислите основные режимы представления документов в MS Word.
2. Какова структура страницы документа Word?
3. Что такое абзац?
4. Какие операции используются при форматировании документа Word в целом?
5. Каким образом нужно завершать работу текстового процессора?
6. Как быстро просмотреть весь документ?
7. Как в документ Word можно вставить другой файл?
8. Создайте таблицу — расписание уроков. Выполните необходимое форматирование. Используя команду *Таблица/Автоформат*, выполните оформление и заливку таблицы.
9. Создайте документ — приглашение. В документ вставьте рисунок. Создайте рамку вокруг рисунка и выполните обтекание текста относительно рисунка.
10. Разработайте визитную карточку.

5.3. ТЕХНОЛОГИЯ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Графические данные в ПК могут использоваться компьютером в двух формах: в виде отдельных битов информации (растровая графика) и в виде минимального набора векторов (векторная графика).

5.3.1. Растровая графика

Растровые графические изображения формируются в процессе преобразования графической информации из аналоговой формы в цифровую (например, в процессе сканирования существующих на бумаге или фотопленке рисунков и фотографий, при использовании цифровых фото- и видеокамер, при просмотре на компьютере телевизионных передач с использованием ТВ-тюнера и т.д.) (см. гл. 3).

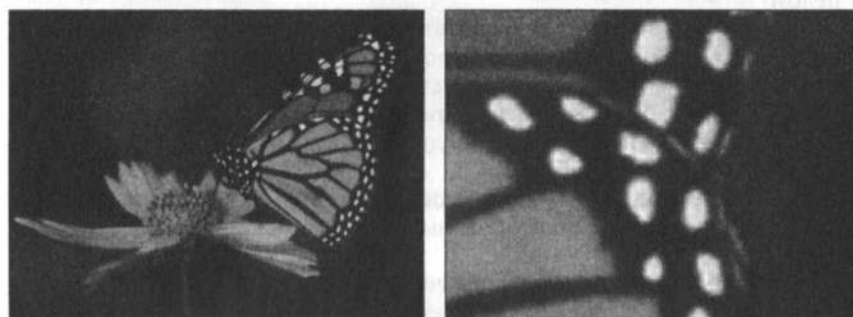
Можно создать растровое графическое изображение и непосредственно на компьютере с использованием графического ре-

дактора, загрузить с дисков CD-ROM или DVD-ROM или «скачать» из Интернета.

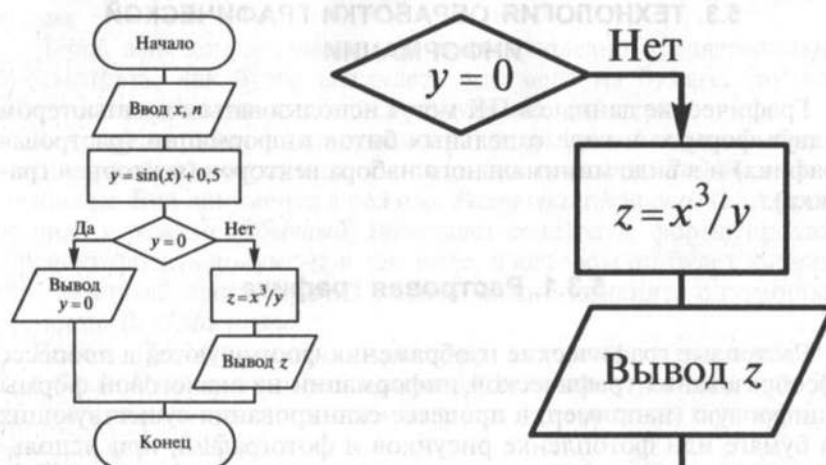
Растровое изображение хранится с помощью точек различного цвета (пикселей), которые образуют строки и столбцы.

Пиксел — минимальный участок изображения, которому независимым образом можно задать цвет. Каждый пиксел имеет определенное положение. Качество растрового изображения зависит от размера изображения (числа пикселей по горизонтали и вертикали) и числа цветов, которые могут принимать пиксели. Хранение каждого пиксела требует определенного числа бит (глубина цвета), которое зависит от числа цветов в изображении.

Такой метод передачи и воспроизведения графических данных эффективен при работе с фотоизображениями, когда необходимо



a



б

Рис. 5.11. Масштабирование изображений:

a — растровое изображение и его увеличенный фрагмент; *б* — векторное изображение и его увеличенный фрагмент

выполнять операции над сканированными графическими данными — фотографиями, рисунками, знаками, картинками, видеоизображениями.

Растровые изображения очень чувствительны к масштабированию (увеличению или уменьшению) (рис. 5.11, а). При уменьшении растрового изображения нескольких соседних точек преобразуются в одну, поэтому теряется разборчивость мелких деталей изображения. При увеличении растрового изображения увеличивается размер каждой точки и появляется ступенчатый эффект, который можно увидеть невооруженным глазом.

5.3.2. Векторная графика

Векторные графические изображения являются оптимальным средством для хранения высокоточных графических объектов (чертежей, схем и т.д.), для которых имеет значение сохранение четких и ясных контуров. Векторные изображения формируются из объектов (точки, линии, окружности, прямоугольника и т.д.), которые хранятся в памяти компьютера в виде графических примитивов и описывающих их математических формул. Например, точка задается своими координатами (X, Y), линия — координатами начала (X_1, Y_1) и конца (X_2, Y_2), окружность — координатами центра (X, Y) и радиусом (R), прямоугольник — длинами сторон и координатами левого верхнего угла (X_1, Y_1) и правого нижнего угла (X_2, Y_2) и т.д. Для каждого примитива задается также цвет.

Векторный метод описания графических данных позволяет выполнять сложные графические изображения и осуществлять различные операции над ними: сжатие, растяжение, разбиение на части, копирование, перемещение и многое другое. Данный метод эффективно сочетается со всевозможными надписями и текстовыми сопровождениями, используется при разработке символики товарных и опознавательных знаков и т.д.

Достоинством векторной графики является то, что файлы, хранящие векторные графические изображения, имеют сравнительно небольшой объем.

Важно также, что векторные графические изображения могут быть увеличены или уменьшены без потери качества (рис. 5.11, б). Это возможно, так как масштабирование изображений производится с помощью простых математических операций (умножения параметров графических примитивов на коэффициент масштабирования).

Векторные изображения не теряют качества при масштабировании и занимают небольшой объем по сравнению с растровыми изображениями.

5.3.3. Форматы графических файлов

Форматы графических файлов определяют способ хранения информации в файле (растровый или векторный), а также форму хранения информации (используемый алгоритм сжатия).

Сжатие применяется для растровых графических файлов, так как они имеют обычно достаточно большой объем. Сжатие графических файлов отличается от их архивации с помощью программ-архиваторов (RAR, ZIP, ARJ, и т.д.) тем, что алгоритм сжатия включается в форматы графического файла.

Некоторые форматы графических файлов являются универсальными, так как могут быть обработаны большинством графических редакторов. Некоторые программы обработки изображений используют оригинальные форматы, которые распознаются только самой создающей программой. Преимущество оригинальных форматов файлов состоит в том, что они позволяют сохранять изображения при меньшем размере файла.

Рассмотрим некоторые форматы графических файлов.

Bit Map image (BMP) — универсальный формат растровых графических файлов, используемый в операционной системе Windows. Этот формат поддерживается многими графическими редакторами и, в том числе, редактором Paint. Рекомендуется для хранения и обмена данными с другими приложениями.

Tagged Image File Format (TIFF) — формат растровых графических файлов, поддерживаемый всеми основными графическими редакторами и компьютерными платформами. Включает в себя алгоритм сжатия без потери информации. Используется для обмена документами между различными программами. Рекомендуется для использования при работе с издательскими системами.

Graphics Interchange Format (GIF) — формат растровых графических файлов, поддерживаемый приложениями для различных операционных систем. Включает в себя алгоритм сжатия без потери информации, позволяющий уменьшить объем файла в несколько раз. Рекомендуется для хранения изображений, создаваемых программным путем (диаграмм, графиков и т.д.) и рисунков (типа аппликации) с ограниченным числом цветов (до 256). Используется для размещения графических изображений на Web-страницах в Интернете.

Portable Network Graphic (PNG) — формат растровых графических файлов, аналогичный формату GIF. Рекомендуется для размещения графических изображений на Web-страницах в Интернете.

Joint Photographic Expert Group (JPEG) — формат растровых графических файлов, который реализует эффективный алгоритм сжатия (метод JPEG) для отсканированных фотографий и иллюстраций. Алгоритм сжатия позволяет уменьшить объем файла в

десятки раз, однако приводит к необратимой потере части информации. Поддерживается приложениями для различных операционных систем. Используется для размещения графических изображений на Web-страницах в Интернете.

Windows MetaFile (WMF) — универсальный формат векторных графических файлов для Windows-приложений. Используется для хранения коллекции графических изображений Microsoft Clip Gallery.

Encapsulated PostScript (EPS) — формат векторных графических файлов, поддерживаемый программами для различных операционных систем. Рекомендуется для печати и создания иллюстраций в настольных издательских системах.

CorelDraw files (CDR) — оригинальный формат векторных графических файлов, используемый в системе обработки векторной графики CorelDraw.

5.3.4. Графические редакторы

Графический редактор — это программа создания, редактирования и просмотра графических изображений.

Графические редакторы можно подразделить на две категории: растровые и векторные.

Растровые графические редакторы (Paintbrush, PhotoShop, Adobe CorelPhotoPaint) основаны на битовом методе передачи изображений. Они обрабатывают достаточно подробные сканируемые образы, выполняя ретуширование, изменение цветов, их оттенков и контрастов, растирание и штриховку, изменение направления тени и контуров. Длина битовых файлов велика из-за большого числа обрабатываемых экранных точек, что позволяет использовать большое число цветов и детально редактировать изображение.

Сканирование образа может быть осуществлено в черно-белом изображении, в серых оттенках и тонах и в цветном изображении. Для растровых графических редакторов необходимо подобрать угол и тип раstra (блочный, линейный, точечный и т.д.), разрешающую способность. Для цветных изображений регулируют яркость и контраст по каждому из цветов, составляющих гамму графического изображения. Готовые рисунки хранятся в отдельных файлах и могут быть доступны для использования.

Векторные графические редакторы (Corel Draw, Pictor, графический редактор Word, Adobe Illustrator) основаны на векторном методе кодирования информации и позволяют в мельчайших подробностях создавать и редактировать вновь создаваемые изображения. Большинство текстовых редакторов, офисных и издательских систем содержат в своем составе графическую программу.

Инструменты рисования объектов. Графические редакторы имеют набор инструментов для рисования простейших графических объектов: прямой линии, кривой, прямоугольника, эллипса, многоугольника и т.д. После выбора объекта на панели инструментов его можно нарисовать в любом месте окна редактора.

Например, для рисования линии необходимо выбрать на панели инструментов инструмент *Линия*, переместить курсор на определенное место окна редактора и щелчком мыши зафиксировать точку, из которой должна начинаться линия. Затем перетащить линию в нужном направлении и, осуществив повторный щелчок, зафиксировать второй край линии.

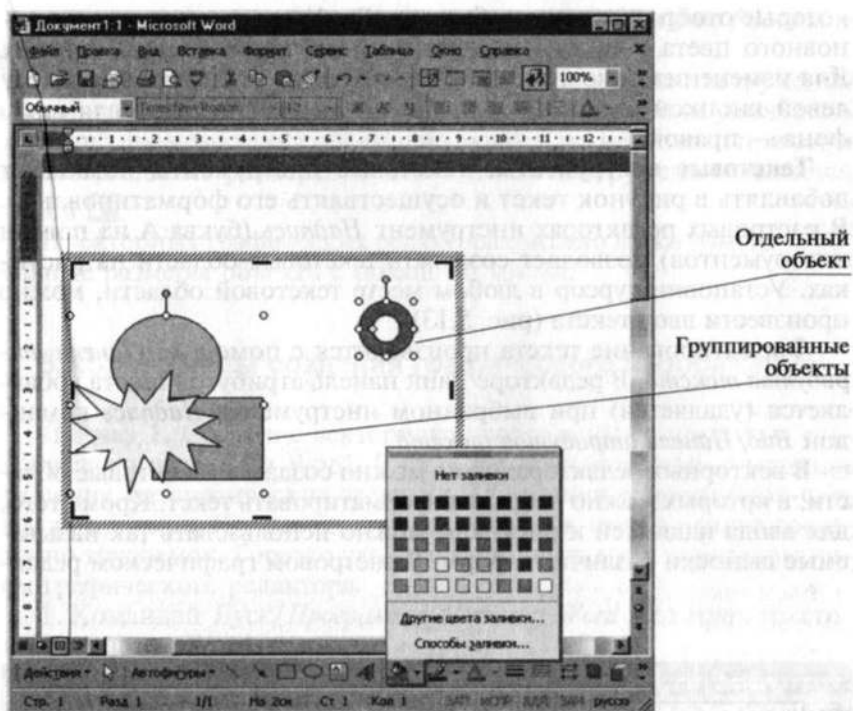
Такие инструменты имеются и в растровом, и векторном графическом редакторах, однако принципы работы с ними несколько различаются. В растровом графическом редакторе этот объект перестает существовать как самостоятельный элемент после окончания рисования и становится лишь группой пикселей на рисунке. В векторном редакторе этот объект продолжает сохранять свою индивидуальность и его можно масштабировать, перемещать по рисунку и т.д.

Выделяющие инструменты. В графических редакторах с элементами изображения можно осуществлять различные операции: копирование, перемещение, удаление, поворот, изменение размеров и т.д. Для того чтобы выполнить какую-либо операцию над объектом, его сначала необходимо выделить.

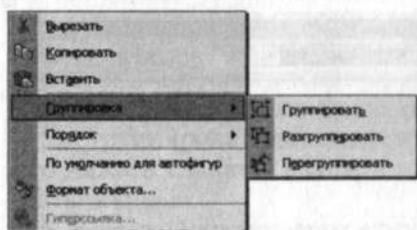
Для выделения объектов в растровом графическом редакторе имеются два инструмента: *Выделение прямоугольной области* и *Выделение произвольной области*. Процедура выделения области производится аналогично процедуре рисования. Выделение объектов в векторном редакторе осуществляется с помощью инструмента *Выделение объекта* (на панели инструментов изображается стрелкой). Для выделения объекта достаточно выбрать инструмент выделения и щелкнуть по любому объекту на рисунке (рис. 5.12, а).

Инструменты группировки и разгруппировки объектов. В графическом редакторе существуют инструменты группировки и разгруппировки объектов, вызов которых осуществляется нажатием правой кнопки мыши на объекте (рис. 5.12, б). Операция группировки объединяет несколько отдельных объектов в один, что позволяет производить над ними общие операции (перемещать, удалять и т.д.), но в этом случае невозможно редактировать отдельный объект. Сгруппированный объект можно разбить на отдельные объекты (разгруппировать).

Инструменты редактирования рисунка. Инструменты редактирования позволяют вносить в рисунок изменения: стирать части рисунка, изменять цвета и т.д. Для стирания изображения в растровых графических редакторах используется инструмент *Ластик*,



а



б

Рис. 5.12. Работа с несколькими объектами:

а — объекты и панель Заливка; б — группировка

который стирает фрагменты изображения, при этом размер *Ластика* можно менять.

В векторных редакторах редактирование изображения возможно только путем удаления объектов, входящих в изображение целиком. Для этого сначала необходимо выделить объект, а затем выполнить операцию *Вырезать*.

Палитра цветов. Операцию изменения цвета можно осуществить с помощью меню *Палитра*, которое содержит набор цветов, используемых при создании объектов. Различают основной цвет, которым рисуются контуры фигур, и цвет фона. В левой части палитры размещаются индикаторы основного цвета и цвета фона,

которые отображают текущие установки (в данном случае для основного цвета установлен черный, а для цвета фона — белый). Для изменения основного фона необходимо осуществить щелчок левой кнопкой мыши на выбранном цвете палитры, а для цвета фона — правой.

Текстовые инструменты. Текстовые инструменты позволяют добавлять в рисунок текст и осуществлять его форматирование. В растровых редакторах инструмент *Надпись* (буква А на панели инструментов) позволяет создавать текстовые области на рисунках. Установив курсор в любом месте текстовой области, можно произвести ввод текста (рис. 5.13).

Форматирование текста производится с помощью *Панели атрибутов текста*. В редакторе Paint панель атрибутов текста добавляется (удаляется) при выбранном инструменте *Надпись* командой *Вид/Панель атрибутов текста*.

В векторных редакторах тоже можно создавать текстовые области, в которых можно вводить и форматировать текст. Кроме того, для ввода надписей к рисункам можно использовать так называемые выноски различных форм. В растровом графическом редак-

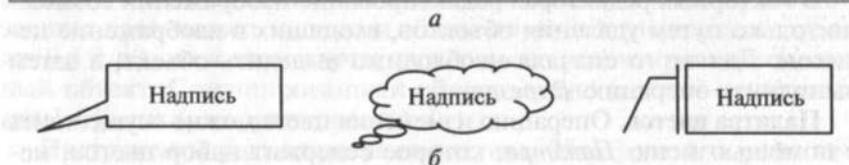
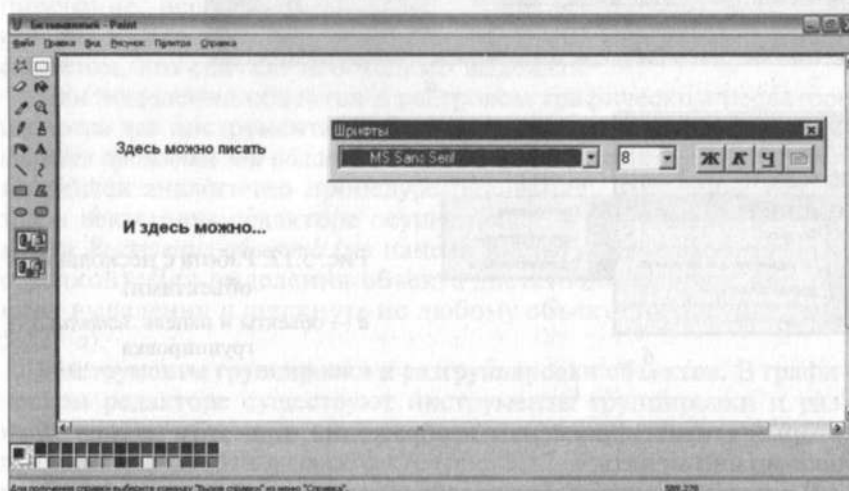


Рис. 5.13. Работа с текстом:

а — при помощи инструмента *Надпись*; б — при помощи инструмента *Выноска*

торе, входящем в Word, выноска выбирается на панели *Рисование* командой *Автофигуры/Выноски* (см. рис. 5.13, б).

Масштабируемые инструменты. Масштабируемые инструменты в графических редакторах позволяют увеличивать или уменьшать масштаб представления объекта на экране, но не влияют при этом на его реальные размеры. Обычно такой инструмент называется лупой .

В векторных графических редакторах можно легко изменять реальные размеры объекта с помощью мыши.

5.3.5. Примеры создания графического изображения

Пример 1. Создание векторного изображения с помощью графического редактора Word. В гл. 7 вы познакомитесь с методом решения математических задач на ПК, использующим блок-схемы алгоритмов. Процесс вычерчивания блок-схем вручную достаточно трудоемок. Гораздо удобнее выполнять его с использованием графического редактора.

1. Командой *Пуск/Программы/Microsoft Word* запустить текстовый редактор Word.

2. Командой *Рисование/пункт «Автофигуры»* выбрать пункт «Блок-схема», содержащий различные элементы блок-схемы.

3. Нарисовать элементы блок-схемы, расположив их в нужном порядке и соединив стрелочками.

4. В контекстных меню каждого из элементов блок-схемы выбрать пункт «Добавить текст» и ввести текст. При необходимости с помощью контекстного меню текста отформатировать текст.

5. Сгруппировать все элементы блок-схемы в один объект, для чего нажать клавишу [Shift] и последовательно выбирать все элементы мышью.

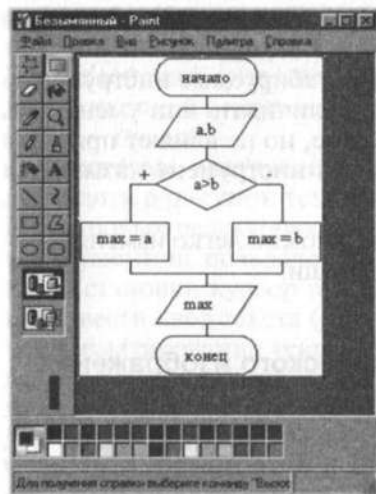
6. В результате получим единый графический объект, который можно с помощью пункта меню *Действия* изменять различными способами: поворачивать, сдвигать и т.д. (рис. 5.14, а).

Графический редактор, входящий в Word, может производить некоторые преобразования и с растровыми изображениями (например, сделать прозрачным фон какой-то части рисунка или вырезать какую-либо его часть).

Пример 2. Создание растрового изображения с помощью графического редактора Paint.

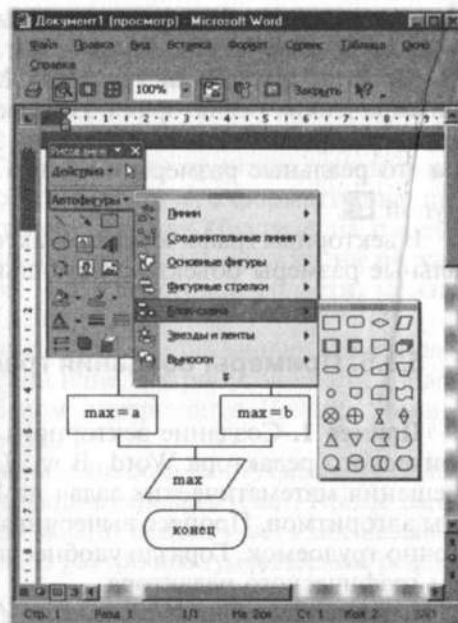
1. Командой *Пуск/Программы/Стандартные/Paint* запустить графический редактор Paint.

2. На панели инструментов нажать кнопку  (*Прямоугольник*), после чего нарисовать в рабочей области экрана прямоугольник.



а

Рис. 5.14. Создание растрового (а) и векторного (б) изображения



б

3. На панели инструментов нажать кнопку (Скругленный прямоугольник), после чего нарисовать в рабочей области экрана скругленный прямоугольник.

4. На панели инструментов нажать кнопку (Надпись), после чего открывается возможность для написания в рабочей области экрана необходимого текста.

5. На панели инструментов нажать кнопку (Линия), после чего нарисовать в рабочей области экрана линии необходимой длины (рис. 5.14, б).

В графических редакторах существуют возможности для редактирования готовых изображений (графических файлов). Для этого необходимо открыть новый документ и командой *Вставить/Рисунок/Из файла* вставить в документ рисунок.

С рисунком можно выполнять некоторые дополнительные действия, например:

- на панели *Рисование* выбрать операцию *Установить прозрачный фон* (кнопка), переместить указатель мыши, принявший форму, изображенную на кнопке, на любую точку фона и щелкнуть. Фон станет прозрачным;
- на панели *Рисование* (Word) выбрать операцию *Обрезка* (кнопка), подвести указатель мыши, принявший форму, изобра-

женную на кнопке, по очереди к маркерам, отмечающим границы рисунка, и сместить их к центру. В результате «отрежутся» ненужные края изображения.

Технология обработки графических файлов в растровых и векторных графических редакторах имеет много общего в виде инструментария и последовательности этапов работы.

Контрольные вопросы

1. В чем различие растровых и векторных графических изображений?
2. Какие форматы графических файлов вы знаете и каковы их особенности?
3. Перечислите свойства изображения, которые следует сохранять в формате GIF, и свойства изображения, которые лучше сохранять в формате JPEG.
4. Как рассчитать объем растрового графического файла, который является копией экрана с разрешением 1000×600 точек и глубиной цвета 32 бит?
5. Какой тип графического изображения (растровый или векторный) вы выберете для разработки символов нового шрифта, учитывая, что шрифт должен масштабироваться без потерь качества изображения?
6. Рисунок имеет размер 300×400 пикселей. Определите объем памяти, необходимый для хранения этого рисунка, если:
 - рисунок черно-белый (без градаций серого цвета);
 - в рисунке используются 256 цветов;
 - рисунок является полноцветным RGB-изображением.
7. Сканируем рисунок размером 3×4 дюйма с разрешением 150 dpi. Определите объем результирующего файла, если:
 - рисунок выполнен в режиме HighColor;
 - в рисунке используются 16 цветов;
 - рисунок выполнен в градациях серого цвета (GrayScale).(Расчеты выполнить с помощью программы *Калькулятор*, результаты представить в виде документа Word).

5.4. ТЕХНОЛОГИЯ ОБРАБОТКИ ЧИСЛОВОЙ ИНФОРМАЦИИ

Простейшим средством обработки числовых данных на компьютере является стандартная программа Windows *Калькулятор*. Однако возможности этой программы ограничены, с ее помощью трудно обрабатывать большие массивы числовых данных, являющихся результатами экспериментов, статистических данных и т.д. В этих случаях используются электронные таблицы, которые позволяют автоматизировать табличные расчеты, редактировать и печатать ведомости, а также наглядно представлять результаты обработки в виде диаграмм и графиков.

Электронная таблица — это работающая в диалоговом режиме программа обработки числовых данных, хранящая и обрабатывающая данные в прямоугольных таблицах.

Наиболее популярными являются: электронная таблица Microsoft Excel, Lotus 1-2-3, QuatroPro. Рассмотрим электронные таблицы на примере Microsoft Excel. Вычислительные возможности Microsoft Excel позволяют создавать любые документы, содержащие текстовые и числовые данные, рисунки, диаграммы.

Табличный процессор Excel используют:

- для решения математических задач (выполнения табличных вычислений, исследования функций решения уравнений);
- решения задач математического моделирования;
- проведения статистического анализа;
- ввода и редактирования текста;
- наглядного представления данных в виде таблиц и функций.

Рассмотрим компоненты окна Excel (рис. 5.15), дополняющие основное окно Windows. Рабочий лист рабочей книги служит для хранения данных, разделенных на ячейки. Строка формул предназначена для редактирования содержимого ячейки. Полосы прокрутки применяются в тех случаях, когда весь текст не помещается на экране и требует сдвига вверх-вниз или влево-вправо. Вкладки

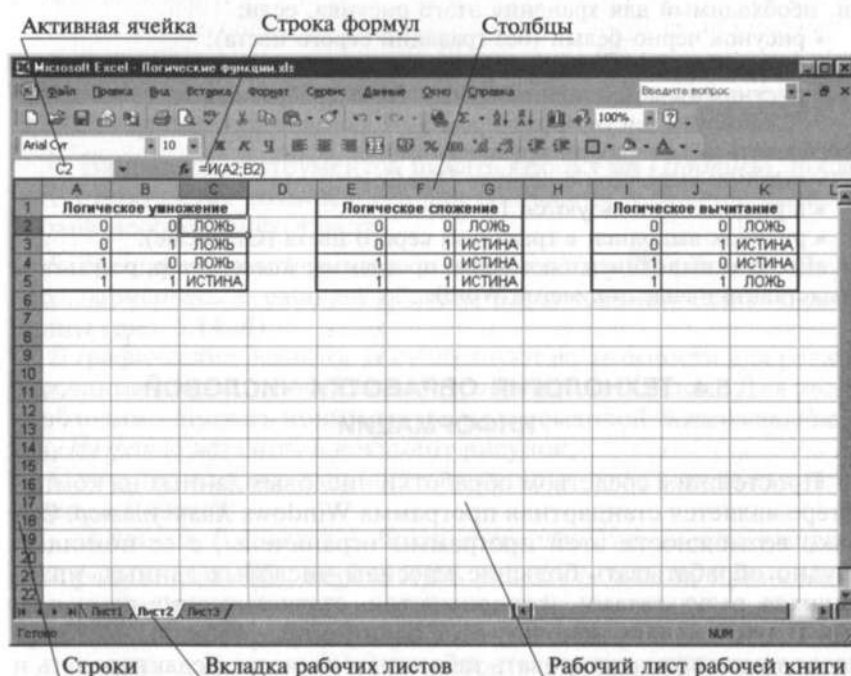


Рис. 5.15. Окно документа Excel

рабочих листов служат для перемещения между рабочими листами книги.

При запуске программы Microsoft Excel появляется рабочая книга. Книга в Microsoft Excel представляет собой файл, используемый для обработки и хранения данных. Каждая книга может состоять из нескольких рабочих листов, поэтому в одном файле можно поместить разнообразные сведения и установить между ними необходимые связи.

Рабочая книга Excel — совокупность рабочих листов, сохраняемых на диске в одном файле. Документом, т. е. объектом обработки Excel, является файл с произвольным именем и расширением *.xls. В каждом файле *.xls может размещаться одна книга, а в книге — от 1 до 255 рабочих листов. По умолчанию в каждой книге содержится три рабочих листа. Рабочий лист имеет табличную структуру и может состоять из любого числа страниц.

Листы служат для организации и анализа данных. Одновременно на нескольких листах данные можно вводить, править, производить с ними вычисления. Имена листов находятся на ярлычках, расположенных в нижней части окна книги. Для перехода с одного листа на другой необходимо указать соответствующий ярлычок. Название текущего листа всегда выделено жирным шрифтом. По умолчанию листы именуются «Лист 1», «Лист 2» и т. п. Листы можно переименовывать, вставлять, удалять, перемещать или копировать в пределах одной книги или из одной книги в другую.

На каждом листе электронной таблицы используется сетка, чтобы разделить лист на строки и столбцы. Заголовки столбцов обозначаются буквами или сочетаниями букв (A, G, AB и т. п.), заголовки строк — числами (1, 16, 278 и т. п.).

Ячейка — область электронной таблицы, находящаяся на месте пересечения столбца и строки и являющаяся наименьшей структурной единицей на рабочем листе. Каждая ячейка таблицы имеет свой собственный адрес. Адрес ячейки электронной таблицы составляется из заголовка столбца и заголовка строки, например: A1, F123, R7.

В Microsoft Excel существует понятие активной (текущей) ячейки. *Активная ячейка* — это ячейка, которая в данный момент способна воспринимать ввод с клавиатуры. Активная ячейка отображается двумя способами:

- 1) в самой таблице (она подсвечена жирной линией), которая называется табличным курсором;
- 2) номер ячейки отображается в строке формул, которая расположена над панелью, содержащей названия столбцов.

Взаимодействие пользователя с программой Microsoft Excel происходит с помощью:

- меню приложений;
- панели инструментов;

- строки формул;
- строки состояния.

Строка формул — строка над рабочей областью, состоящая из трех частей. Первая часть содержит адрес текущей ячейки. При вводе данных в текущую ячейку они автоматически отображаются в третьей части строки формул. В средней части находятся кнопки, появляющиеся при вводе или редактировании данных.

Строка состояния — область в нижней части окна, отображающая сведения о выбранной команде или выполняемой операции.

5.4.1. Типы данных, используемых в Microsoft Excel

В Microsoft Excel поддерживаются следующие типы данных: текст; числа; даты; формулы; функции.

Текст — любая последовательность, состоящая из цифр, пробелов и нецифровых символов. Введенный текст выравнивается в ячейке по левому краю.

Числа могут включать в себя цифры и различные символы: знак процента, знак мантиссы, круглые скобки, денежные обозначения, разделители и др.

Дата и время вводятся в ячейке как числа и выравниваются по правому краю. Программа Excel распознает внешние форматы дат, в которых день, месяц и год разделены точкой, косой чертой или пробелом. Например: 10.12.99, 10/12/99 или 10 декабря 1999.

Формулы представляют собой последовательность значений, ссылок на ячейки, имен, функций или операторов и вычисляют новое значение на основе существующих. Формула всегда начинается со знака равенства (=).

Например, формула $=A1+B2$ обеспечивает сложение чисел, хранящихся в ячейках A1 и B2, а формула $=A1*9$ — умножение числа, хранящегося в ячейке A1, на 9. При вводе формулы в ячейке отображается не сама формула, а результат вычислений по этой формуле. При изменении исходных значений, входящих в формулу, результат пересчитывается немедленно.

В зависимости от решаемой задачи возникает необходимость применять различные форматы представления данных. В каждом конкретном случае важно выбрать наиболее подходящий формат.

По умолчанию Excel использует для представления чисел «Числовой» формат, который по умолчанию отображает два десятичных знака после запятой (например, 195,20).

Экспоненциальный формат применяется, если число, содержащее большое число разрядов, не помещается в ячейке (например, число 2 000 000 000 в экспоненциальном формате будет записано в следующем виде: 2,00E + 09). Можно использовать также специализированные форматы: «Денежный» формат удобен для бухгал-

терских расчетов, форматы «Дата» и «Время» позволяют хранить значение временных данных, «Текстовый» формат предназначен для хранения текста и т.д.

Для выбора формата данных необходимо выбрать команду *Формат/Формат ячеек*. В диалоговом окне *Формат ячеек* выбрать вкладку *Число*. В списке «Числовые форматы» выбрать необходимый формат, например «Числовой». С помощью счетчика *Число десятичных знаков* установить необходимое число знаков после запятой. В области *Отрицательные числа* выбрать форму представления отрицательных чисел.

Перед вводом данных необходимо правильно установить формат ячеек.

5.4.2. Относительные и абсолютные ссылки

В формулах используются ссылки на адреса ячеек. Существуют два основных типа ссылок: относительные и абсолютные. Различия между относительными и абсолютными ссылками проявляются при копировании формулы из активной ячейки в другую ячейку. Относительная ссылка в формуле используется для указания адреса ячейки, вычисляемого относительно ячейки, в которой находится формула. При перемещении или копировании формулы из активной ячейки относительные ссылки автоматически обновляются в зависимости от нового положения формулы. Относительные ссылки имеют следующий вид: A1, B3. По умолчанию при наборе формул в Excel используются относительные ссылки. Абсолютная ссылка в формуле используется для указания фиксированного адреса ячейки.

При перемещении или копировании формулы абсолютные ссылки не изменяются. В абсолютных ссылках перед неизменяемым значением адреса ячейки ставится знак доллара (например: \$A\$1).

Рассмотрим действие относительных и абсолютных ссылок на примере. Пусть нам необходимо вычислить стоимость комплектующих для компьютера в рублях, если известны их цены в долларах и курс рубля. Для вычисления цены в рублях необходимо умножить цену в долларах на курс доллара. Ниже приведена технология работы (рис. 5.16, а).

1. Ввести в ячейки A5, A6 и A7 названия устройств, а в ячейки B5, B6, B7 — их цены в долларах.
2. Ввести в ячейку C2 курс рубля.
3. Ввести в ячейку C5 формулу $=B5*\$C\2 , где B5 — относительная ссылка, а \$C\$2 — абсолютная.
4. Скопировать формулы в ячейки C6 и C7; абсолютная ссылка на ячейку \$C\$2 останется неизменной, а относительная B5 изменится на величину смещения от активной ячейки.

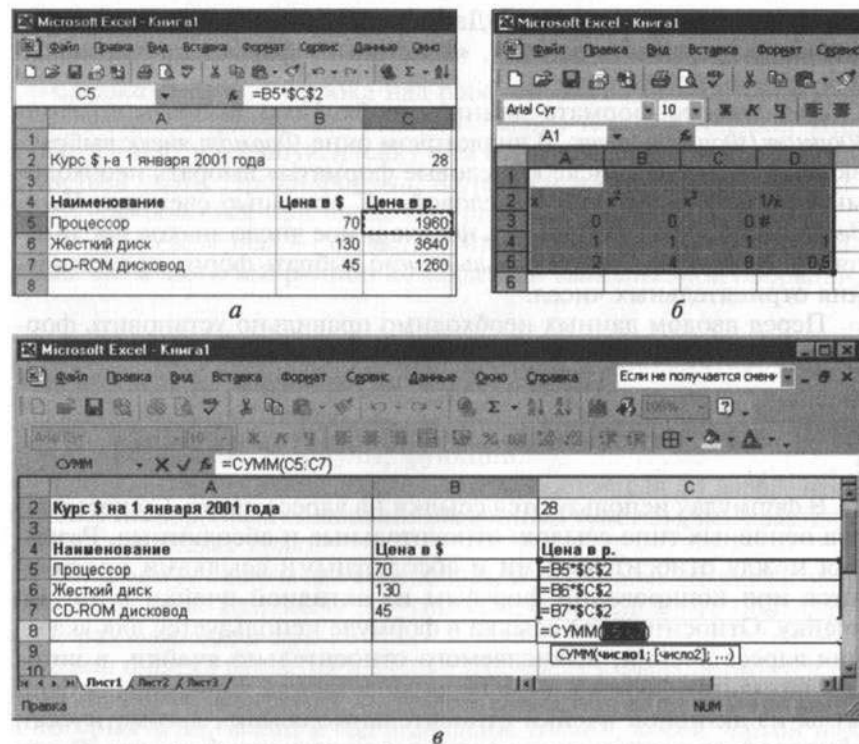


Рис. 5.16. Технология работы с данными:

а — использование относительных и абсолютных ссылок; б — выделение диапазона ячеек; в — автосуммирование

Если символ доллара стоит перед буквой (например: \$A1), то координата столбца абсолютная, а строки — относительная. Если символ доллара стоит перед числом (например: A\$1), то, наоборот, координата столбца относительная, а строки — абсолютная. Такие ссылки называются смешанными.

Для указания фиксированного адреса ячейки используется абсолютная ссылка, а при возможности перемещения или копирования — относительная.

5.4.3. Табличные вычисления

Возможность использования формул и функций является одним из важнейших свойств табличного процессора. Это, в частности, позволяет проводить статистический анализ числовых значений в таблице. В качестве аргументов в формуле обычно использу-

ются числа и адреса ячеек. Для обозначения арифметических операций могут использоваться следующие символы: «+» (сложение); «—» (вычитание); «*» (умножение); «/» (деление).

Формула может содержать ссылки на ячейки, которые расположены на другом рабочем листе или даже в таблице другого файла. Однажды введенная формула может быть в любое время модифицирована. Встроенный *Менеджер формул* помогает пользователю найти ошибку или неправильную ссылку в большой таблице.

Кроме того, программа Excel позволяет работать со сложными формулами, содержащими несколько операций. Для наглядности можно включить текстовый режим, тогда программа Excel будет выводить в ячейку не результат вычисления формулы, а саму формулу.

Предположим, что в ячейке A1 таблицы находится число 100, а в ячейке B1 — число 20. Чтобы разделить первое число на второе и результат поместить в ячейку C1, в ячейку C1 следует ввести соответствующую формулу ($=A1/B1$) и нажать клавишу [Enter]. Ввод формул можно существенно упростить, используя маленький трюк. После ввода знака равенства следует просто щелкнуть мышью по первой ячейке, затем ввести операцию деления и щелкнуть по второй ячейке (рис. 5.17, а).

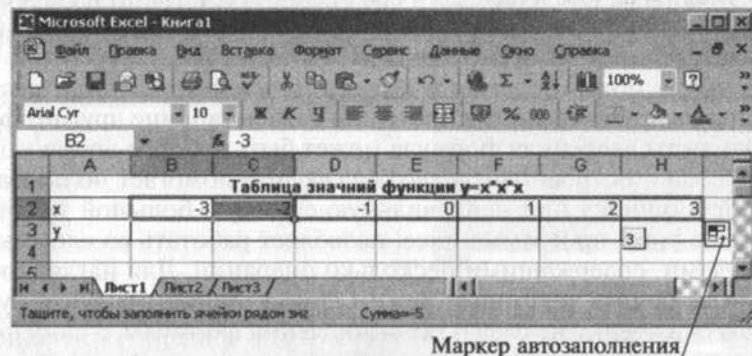
Диапазон — это группа выбранных ячеек. Для выбора прямоугольной области необходимо перевести указатель мыши в один из углов выбираемой области и, удерживая левую кнопку мыши, протянуть указатель в противоположный угол области (рис. 5.17, б). После отпускания кнопки мыши все ячейки, кроме первой, выделяются инверсным цветом. Для выбора целых строк и столбцов можно использовать маркеры строк и столбцов по краям рабочей области. Щелчок на кнопке в верхнем левом углу области позволяет выбрать весь рабочий лист целиком. Если при выборе ячеек удерживать нажатой клавишу [Ctrl], то можно добавлять новые диапазоны к уже выбранному. Этим приемом можно создавать несвязанные диапазоны.

С выбранным диапазоном можно работать, как и с выбранным блоком в программе Word, т. е. можно осуществлять форматирование, копирование, вставку, удаление, перемещение, заполнение, очистку форматов, содержимого, значений.

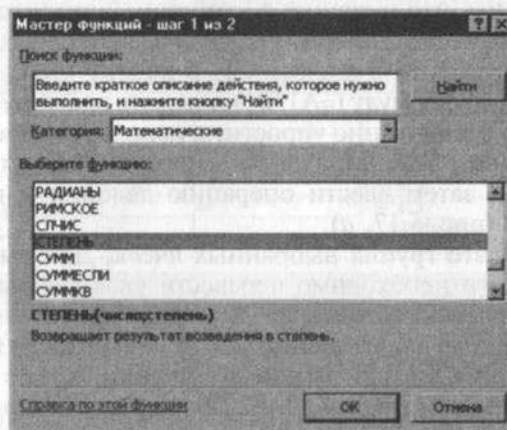
Однако жесткие табличные структуры вносят свои ограничения и дополнительные особенности.

Нажатие клавиши [Del] приводит к удалению содержимого выбранных ячеек. Удаление самих ячеек сопровождается изменением структуры таблицы, т. е. происходит смещение ячеек, новые ячейки становятся на место удаленных.

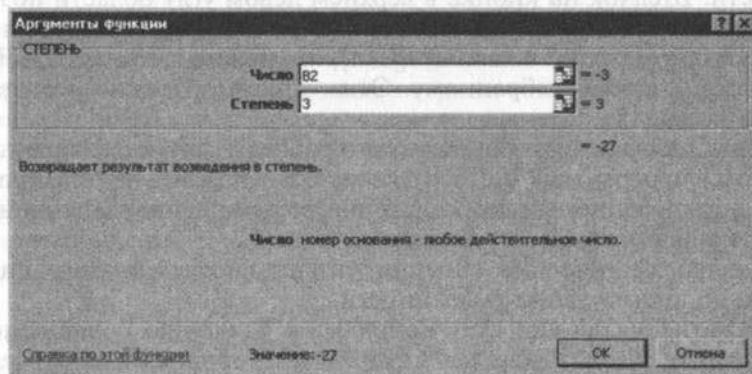
При копировании и вырезании ячеек выбранный диапазон обводится пунктирной рамкой. Копировать и перемещать ячейки можно либо с помощью буфера обмена, либо перетаскиванием.



a



б



в

Рис. 5.17. Использование Мастера функций:

a — автоматизация ввода данных; б — диалоговое окно Мастера функций: шаг 1; в — диалоговое окно Мастера функций: шаг 2

Для этого надо установить указатель мыши на границу текущей ячейки или выбранного диапазона и после того как он примет вид стрелки, можно производить перетаскивание. Если при перетаскивании использовать правую кнопку мыши, то после отпущения откроется специальное меню, позволяющее выбрать произвольную операцию.

5.4.4. Встроенные функции

Формулы могут состоять не только из арифметических операторов и адресов ячеек. Часто в вычислениях приходится использовать формулы, содержащие функции. Excel имеет несколько сотен функций, которые подразделяются на следующие группы: «Математические», «Логические», «Статистические», «Финансовые», «Дата и время» и т.д. Рассмотрим на примерах работу с математическими и логическими функциями.

Математические функции. Одной из наиболее часто используемых операций является суммирование значений диапазона ячеек для расчета итоговых результатов. На панели инструментов *Стандартная* расположена кнопка  (*Автосуммирование*), которая используется для автоматического суммирования чисел с помощью функции СУММ.

Пример 1. Таблица содержит цены на комплектующие компьютера. Вычислить их суммарную стоимость (рис. 5.16, в).

Произвести суммирование значений диапазона ячеек.

1. Выделить ячейку C8, в которую следует поместить сумму.
2. Нажать кнопку , после чего будет выделен диапазон ячеек для суммирования СУММ (C5:C7).
3. Если предложенный диапазон не подходит, перетащить указатель по ячейкам, которые нужно просуммировать. Нажать клавишу [Enter].

Пример 2. Составить таблицу значений функции $y = x^3$ на отрезке $[-3; 3]$ с шагом 1.

Для составления таблицы значений функции можно воспользоваться *Мастером функций*.

1. Подготовить таблицу, ввести значения аргумента. Выделить ячейку, в которую нужно вставить первое значение функции.

Можно автоматизировать процесс заполнения данных, используя маркер заполнения. Для того чтобы воспользоваться маркером заполнения, необходимо установить указатель мыши на правый нижний угол рамки текущей ячейки (рис. 5.17, а). Указатель мыши примет форму крестика. Перетаскивание маркера заполнения позволяет размножить содержимое текущей ячейки на несколько ячеек в строке или столбце.

2. Ввести команду *Вставка/Функция*.
3. В окне диалога *Мастер функции: шаг 1* в списке «Категории» выбрать вариант «Математические», а в списке «Функции» выбрать вариант «Степень». Нажать кнопку *ОК* (рис. 5.17, б).
4. Ввести значения числа и значения показателя степени. Чтобы ввести значение числа, достаточно щелкнуть по нужной ячейке таблицы (рис. 5.17, в).
5. Заполнить ряд значений функции с помощью команды *Правка/Заполнить/Вправо*.

Логические функции. Ранее мы рассмотрели базовые логические операции (умножения, сложения, отрицания) и их таблицы истинности. В электронных таблицах имеются соответствующие логические функции, с помощью которых достаточно просто построить таблицы истинности логических операций.

Аргументами логических функций являются логические значения **ИСТИНА** (1) или **ЛОЖЬ** (0). Логические значения, в свою очередь, могут быть получены как результат определения значения логических выражений. Например, для логического выражения $10 > 5$ результатом будет логическое значение **ИСТИНА**, а для логического выражения $A1 < A2$ (где в ячейке $A1$ хранится число 10, а в ячейке $A2$ — число 5) — логическое значение **ЛОЖЬ**.

Логическая функция **И** имеет в качестве аргументов логические значения, которые могут быть истинными или ложными, и задается формулой $=И(лог_знач1; лог_знач2; ...)$. Функция принимает значение **ИСТИНА** только тогда, когда все аргументы имеют значение **ИСТИНА**. Например, значение функции $=И(10 > 5; 10 < 5)$ — **ЛОЖЬ**.

Логическая функция **ИЛИ** имеет в качестве аргументов логические значения и задается формулой $=ИЛИ(лог_знач1; лог_знач2; ...)$. Функция принимает значение **ИСТИНА**, если хотя бы один из аргументов имеет значение **ИСТИНА**. Например, значение функции $=ИЛИ(10 > 5; 10 < 5)$ — **ИСТИНА**.

Логическая функция **НЕ** меняет значение своего аргумента на противоположное и задается формулой $=НЕ(лог_знач1)$. Функция принимает значение **ИСТИНА**, если аргумент имеет значение **ЛОЖЬ**, и наоборот. Например, значение функции $=НЕ(10 > 5)$ — **ЛОЖЬ**.

Построим с помощью электронных таблиц таблицу истинности операции логического умножения, используя логическую функцию **И**.

Пример 3. Построение таблицы истинности логических операций (рис. 5.18).

1. В пары ячеек ($A2, B2$), ($A3, B3$), ($A4, B4$) и ($A5, B5$) ввести пары значений аргументов логической операции (0,0), (0,1), (1,0) и (1,1).

	A	B	C	D	E	F	G	H	I	J	K
1	Логическое умножение				Логическое сложение				Логическое вычитание		
2	0	0	ЛОЖЬ		0	0	ЛОЖЬ		0	0	ЛОЖЬ
3	0	1	ЛОЖЬ		0	1	ИСТИНА		0	1	ИСТИНА
4	1	0	ЛОЖЬ		1	0	ИСТИНА		1	0	ИСТИНА
5	1	1	ИСТИНА		1	1	ИСТИНА		1	1	ЛОЖЬ

Рис. 5.18. Пример использования логических функций

2. В ячейку C2 ввести формулу логического умножения: =И(A2;B2).
3. Скопировать формулу в ячейки C3, C4 и C5.
4. Значением этой функции в трех случаях является ЛОЖЬ и только в последнем — ИСТИНА.

Повторите шаги 2, 3 для операций логического сложения и отрицания.

Таким образом, в результате мы получим таблицу истинности логических операций.

5.4.5. Сортировка и поиск данных

Электронные таблицы позволяют осуществлять сортировку данных, т. е. производить их упорядочение. Данные (числа, текст, даты, логические значения) в электронных таблицах можно сортировать по возрастанию или убыванию. При сортировке данные выстраиваются в определенном порядке:

- числа сортируются от наименьшего отрицательного до наибольшего положительного числа;
- текст сортируется в следующем порядке: числа, знаки, латинский алфавит, русский алфавит;
- логическое значение ЛОЖЬ предшествует логическому значению ИСТИНА;
- все ошибочные значения равны;
- пустые ячейки всегда помещаются в конец списка.

В процессе сортировки строк таблицы необходимо выбрать столбец, данные которого упорядочиваются. После сортировки изменяется порядок следования строк, но сохраняется их целостность.

Можно проводить вложенные сортировки, т. е. сортировать данные по нескольким столбцам; при этом задается последовательность сортировки столбцов. Если в столбце, по которому осуществляется сортировка, имеются одинаковые значения, то можно задать сортировку по второму столбцу (вложенную сортировку).

Microsoft Excel - Книга1

Исходные данные	А	В	С	Д	Е	Ж	З	И	К	Л	М	Н
1	Наименование устройства	Цена (в у.е.)										
2	Мышь	5										
3	Дисковод 3,5"	14										
4	Корпус	25										
5	Видеоплата 8Мб	30										
6	Клавиатура	10										
7	Звуковая карта 16 бит	30										
8	Монитор 15"	200										
9	Память 32 Мб	45										
10	Дисковод CD-ROM x44	50										
11	Жесткий диск 10 Гб	130										
12	Системная плата	80										
13	Процессор Celeron (400 Mhz)											
14												

а

Сортировка диапазона

Сортировать по:

Наименование устройств (не сортировать)

Цена (в у.е.)

В порядке возрастания / убывания:

Идентифицировать поля по:

☒ подписан (первая строка диапазона)

☐ обозначенный столбцом листа

Параметры... ОК Отмена

б

Microsoft Excel - пример про СММ

	А	В
1	Наименование устройства	Цена (в у.е.)
2	Мышь	5
3	Клавиатура	10
4	Дисковод 3,5"	14
5	Корпус	25
6	Видеоплата 8 Мб	30
7	Звуковая карта 16 бит	30
8	Память 32 Мб	45
9	Дисковод CD-ROM x44	50
10	Процессор Celeron (400 Mhz)	70
11	Системная плата	80
12	Жесткий диск 10 Гб	130
13	Монитор 15"	200
14		

в

Рис. 5.19. Сортировка данных:

а — таблица с исходными данными;
б — диалоговое окно *Сортировка диапазона*; в — результат сортировки

В этом случае ячейки, имеющие одинаковые значения по первому столбцу, будут упорядочены по второму.

На рис. 5.19, а представлена таблица цен на комплектующие компьютера. На этом примере рассмотрим операцию упорядочения данных по столбцу, содержащему цену устройств.

Выделить одну из ячеек с данными и ввести команду *Данные/Сортировка*.

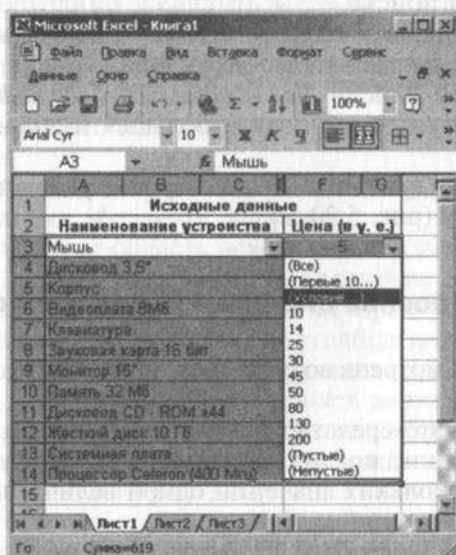
На диалоговой панели *Сортировка диапазона* (рис. 5.19, б) в списке *Сортировать по* выбрать название нужного столбца (в данном случае *Цена*) и установить переключатель в положении по возрастанию.

После нажатия кнопки *ОК* мы получим отсортированную таблицу, в которой устройства расположены в порядке возрастания их цены (рис. 5.19, в).

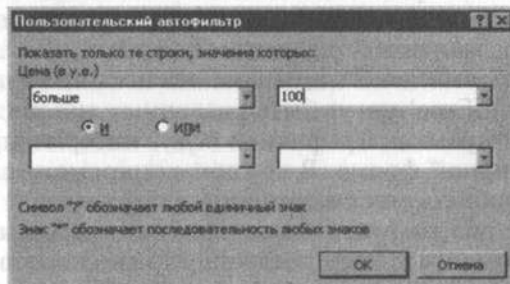
Перед выполнением вложенных сортировок необходимо правильно задать последовательность сортировки столбцов.

В электронных таблицах можно осуществлять поиск данных (строк) в соответствии с заданными условиями. Такие условия

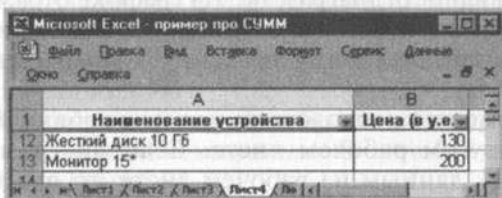
называются фильтром. В результате поиска будут найдены те строки, которые удовлетворяют заданному фильтру. Прежде чем начать поиск, необходимо выделить хотя бы одну ячейку с данными.



а



б



в

Рис. 5.20. Автофильтр:

а — диалоговое окно Автофильтр; б — диалоговое окно Пользовательский автофильтр; в — результат применения Автофильтра

Например, мы хотим найти все строки, которые содержат информацию об устройствах стоимостью более 100 у.е.

Ввести команду *Данные/Фильтр/Автофильтр*. В названиях столбцов таблицы появятся раскрывающиеся списки, содержащие стандартные условия поиска.

Развернуть список в столбце «Цена» и выбрать пункт «Условие...» для ввода пользовательских условий (рис. 5.20, а).

На диалоговой панели *Пользовательский автофильтр* в полях ввести оператор условия поиска «больше» и значение 100 (рис. 5.20, б).

В результате будут найдены две строки, удовлетворяющие заданному фильтру (рис. 5.20, в).

5.4.6. Построение диаграмм и графиков функций

В Excel предусмотрена возможность представлять данные в графическом виде.

Диаграмма — это средство графического представления количества информации, предназначенное для сравнения значений величин или нескольких значений одной величины, слежения за изменением их значений и т.д.

Как правило, диаграммы строятся в прямоугольной системе координат, где по горизонтальной оси ОХ откладываются значения независимой переменной (аргумента), а по вертикальной оси ОУ — значения зависимой переменной (функции). На один рисунок может быть выведено одновременно несколько диаграмм.

Диаграммы в Excel могут быть различных типов (линейчатые, круговые, гистограммы, графики и т.д.), которые представляют данные в различной форме. В каждом конкретном случае важно правильно подобрать тип создаваемой диаграммы.

Столбчатые диаграммы называются гистограммами и отражают дискретное распределение величин. Можно построить как одиночную, так и множественную гистограмму. График рассматривается как разновидность диаграммы. На графике отражается непрерывный процесс изменения величины. Круговая диаграмма используется для представления вклада нескольких величин в единое целое.

Диаграмму можно расположить рядом с таблицей или разместить ее на другом рабочем листе. Если диаграмма является иллюстрацией к данным на рабочем листе, то она должна быть отображена на одном листе с ними (внедренная диаграмма). Если диаграмма будет использоваться самостоятельно, то ее целесообразно разместить на отдельном листе рабочей книги. Листы с диаграммами получают названия «Диаграмма1», «Диаграмма2» и т.д.

Мастер диаграмм является одним из наиболее мощных средств в программе Excel. Построение диаграммы с его помощью выполняется за несколько шагов. На основной панели имеется пиктограмма для вызова *Мастера диаграмм* . Прежде чем строить диаграмму, необходимо закончить все работы в таблице, включая ее форматирование. Если вносятся изменения в ту часть таблицы, по которой строилась диаграмма, то программа Excel автоматически модифицирует диаграмму.

Построение любого вида диаграммы осуществляется за четыре шага.

- Шаг 1 позволяет выбрать тип диаграммы из широкого диапазона предложенных *Мастером диаграмм*.

- Шаг 2 определяет область исходных данных, для которой будет строиться диаграмма, а также направление размещения данных по столбцам или по строкам. При выборе по столбцам *x*-координаты берутся из крайнего левого столбца выделенного блока. Остальные столбцы содержат *y*-координаты диаграмм. По числу столбцов определяется число строящихся диаграмм. При выборе по строкам самая верхняя строка выделенного блока является строкой *x*-координат, остальные строки содержат *y*-координаты.

- Шаг 3 позволяет настроить оформление диаграммы: добавить заголовки по осям и общий заголовок диаграммы, легенду — прямоугольную таблицу, содержащую для каждого ряда данных *ключ* и поле «Название ряда». Ключ легенды повторяет цвет и узор, заданный для элементов данных ряда. Легенда может создаваться в автоматическом режиме, если при построении диаграммы интервал ячеек включал названия рядов. Если имена рядов явно не заданы, то легенда использует стандартное имя *Ряд (номер)*. Легенду можно разместить в различных областях диаграммы (внизу, вверху, слева, справа). Кроме того, на шаге 3 можно добавить на область диаграммы таблицу с данными. Это наиболее удобно, если диаграмма размещается на отдельном листе.

- Шаг 4 определяет месторасположение диаграммы в рабочей книге: на месте с данными или на отдельном листе.

Рассмотрим построение диаграммы на примере таблицы, представленной на рис. 5.21, а.

1. Запустить *Мастер диаграмм* с помощью команды *Вставка/Диаграмма* или кнопки .

2. На первом шаге необходимо выбрать тип диаграммы. В списке «Тип» выбираем пункт «Гистограмма». В окне *Вид* выбираем обычную гистограмму (рис. 5.21, б).

3. На втором шаге мы увидим, как будет выглядеть наша диаграмма. Справа от диаграммы появляется *Легенда*, которая содержит необходимые пояснения к диаграмме. Если в исходной таблице не выделить столбец с названиями магазинов, то в *Легенде* появятся стандартные надписи «Ряд 1», «Ряд 2», «Ряд 3». Окно

Microsoft Excel - Книга1

Файл Правка Вид Вставка Формат Сервис Данные Справка

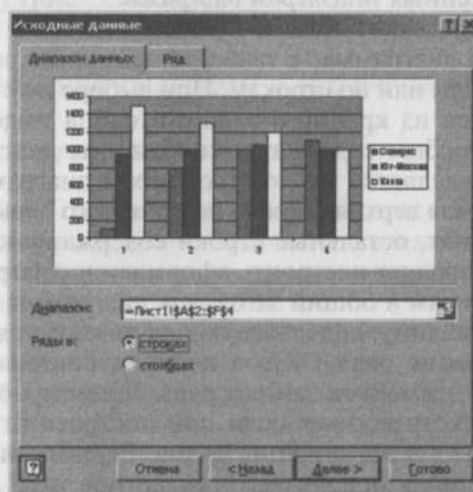
Справка

Arial Cyr 10

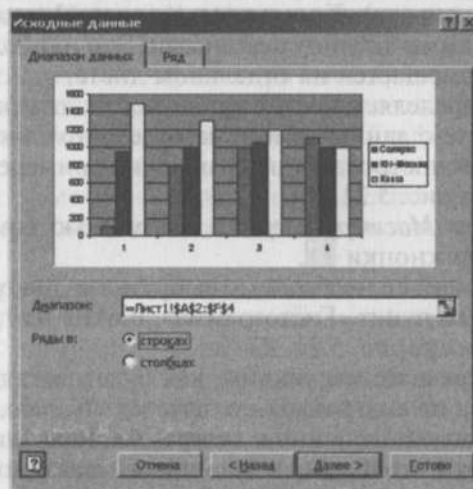
	A	B	C	D	E	F	G
1	Магазины	1 квартал	2 квартал	3 квартал	4 квартал		
2	Соларис	100	800	1000	1100		
3	Юг-Москва	950	1000	1050	1000		
4	Кинга	1500	1300	1200	1000		
5							

Готово

a



б



б

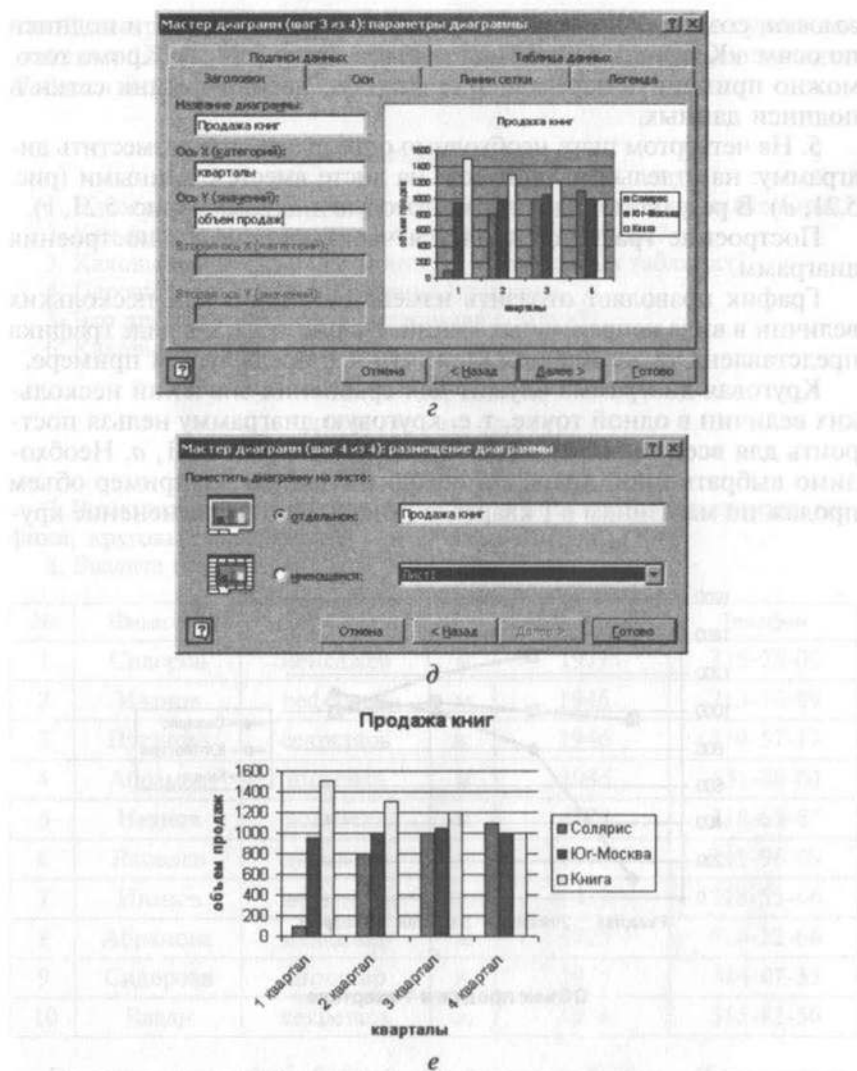


Рис. 5.21. Этапы построения диаграммы:

а — таблица с исходными данными; б — шаг 1 Мастера диаграмм (выбор типа диаграммы); в — шаг 2 Мастера диаграмм (ввод исходных данных); г — шаг 3 Мастера диаграмм (параметры); д — шаг 4 Мастера диаграмм (размещение); е — готовая диаграмма

Диапазон содержит диапазон адресов ячеек, содержащих данные для диаграммы. Этот диапазон можно изменить (рис. 5.21, в).

4. На третьем шаге мы можем уточнить детали отображения диаграммы в окне *Параметры диаграммы*. Используя вкладку *За-*

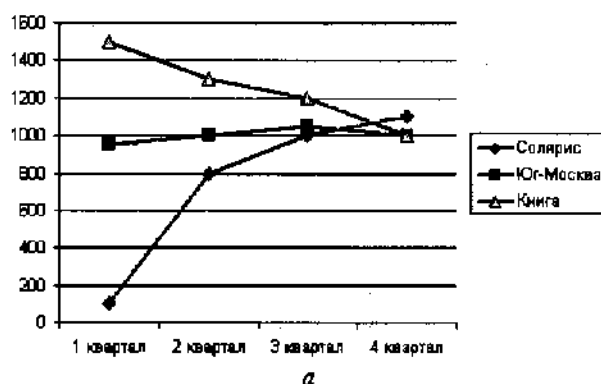
головки, создаем название диаграммы «Продажа книг» и подписи по осям: «Кварталы» и «Объем продаж» (рис. 5.21, г). Кроме того, можно применить перемещение *Легенды*, добавить мини сетки и подписи данных.

5. На четвертом шаге необходимо определить, где разместить диаграмму: на отдельном листе или на листе вместе с данными (рис. 5.21, д). В результате мы получим готовую диаграмму (рис. 5.21, е).

Построение графиков является частным случаем построения диаграмм.

График позволяет отразить изменение одной или нескольких величин в виде непрерывных линий. На рис. 5.22, а в виде графика представлена та же информация, что и в предыдущем примере.

Круговая диаграмма служит для сравнения значений нескольких величин в одной точке, т. е. круговую диаграмму нельзя построить для всей таблицы, представленной на рис. 5.21, а. Необходимо выбрать иной диапазон исходных данных, например объем продаж по магазинам в I квартале (рис. 5.22, б). Применение кру-



Объем продаж в 1 квартале



б

Рис. 5.22. Примеры диаграмм:
а — график; б — круговая

говой диаграммы особенно наглядно, если величины в сумме составляют 100 %. Это осуществляется на шаге 3 с помощью вкладки *Подписи данных* установкой флажка «Доли».

Контрольные вопросы

1. В чем заключается отличие абсолютных ссылок от относительных?
2. Как задается имя ячейки в электронных таблицах?
3. Каковы основные типы данных в электронных таблицах?
4. Перечислите этапы построения диаграмм.
5. Что представляет собой смешанная ссылка?
6. Каковы значения следующих логических функций:

=И (20 > 15; 20 < 15);

=ИЛИ (20 > 15; 20 < 15);

=НЕ (20 > 15)?

7. В каких ситуациях предпочтительнее использовать гистограммы, графики, круговые диаграммы?

8. Введите таблицу:

№	Фамилия	Должность	Пол	Год рождения	Телефон
1	Сидоров	менеджер	м	1977	315-75-09
2	Иванов	референт	м	1946	213-76-89
3	Путачева	секретарь	ж	1946	314-57-17
4	Абрамов	инженер	м	1945	551-88-00
5	Иванов	водитель	м	1978	218-65-87
6	Яковлев	директор	м	1973	211-96-09
7	Иванов	водитель	м	1930	218-55-66
8	Абрамова	менеджер	ж	1925	510-22-66
9	Сидорова	инженер	ж	1955	314-07-33
10	Влади	секретарь	ж	1936	315-82-56

Пользуясь командами *Данные — Сортировка* и *Данные — Итоги*, ответьте на вопросы а...е. Укажите, сколько уровней сортировки нужно использовать, название полей, по которым производится сортировка, и направление сортировки по каждому уровню. Предварительно скопируйте исходную таблицу на отдельные листы или используйте режим групповой работы с листами (результат выполнения каждого пункта должен располагаться на отдельном листе).

- а) Есть ли в фирме женщины-секретари?
- б) Сколько Ивановых работают в фирме и кто из них самый молодой?
- в) Каков средний возраст мужчин и женщин, работающих в фирме?
- г) Сколько в фирме менеджеров, инженеров, водителей и представителей других должностей?

е) Вы — бухгалтер фирмы и должны к 8 Марта выписать премии женщинам. Размер премии зависит от должности и возраста дамы.

9. Сортировка списка по столбцам.

9.1. Отсортируйте таблицу так, чтобы фамилии располагались в алфавитном порядке.

9.2. Скопируйте таблицу на свободный лист. На этом листе транспонируйте таблицу (*Правка => Специальная вставка*). Добавьте в конец таблицы строку «Средний возраст» и заполните ее, введя соответствующую формулу.

9.3. Выполните сортировки по столбцам. Перед каждой сортировкой копируйте таблицу на новый лист.

9.4. Переставьте столбцы так, чтобы фамилии расположились в обратном алфавитном порядке.

10. Изменение структуры таблицы.

10.1. Введите дополнительное поле «Оклад» после поля «Пол» и заполните его осмысленными значениями.

10.2. После поля «Оклад» добавьте еще три поля: «Надбавки», «Налоги», «К выплате».

10.3. Установите надбавки в размере 100 р. женщинам старше 50 лет и мужчинам старше 60 лет. Директору и референту — 200 р.

10.4. В обычном режиме редактирования заполните поле «Налог» — 5 % от суммы оклада и надбавки, если она не превышает 1 тыс. р., и 10 %, если превышает. Используйте функцию ЕСЛИ.

10.5. Запишите формулу и заполните поле «К выплате» («Оклад» + «Надбавки» — «Налоги»).

11. Выполните поиск сотрудников по следующим критериям:

- женщины с низким окладом (меньше 400 р.);
- мужчины, получающие больше 1 тыс. р. и моложе 30 лет;
- женщины, имеющие надбавки и проживающие в Центральном районе, — номера телефонов начинаются на 31.

12. Фильтрация записей с помощью функции автофильтра.

12.1. Для выполнения задания скопируйте итоговый список сотрудников в новую книгу.

12.2. Установите автофильтр (меню *Данные — Фильтр — Автофильтр*) и, выбирая соответствующие должности, примите на работу новых сотрудников. Штат должен содержать трех менеджеров, четырех инженеров, двух секретарей, двух референтов, одного водителя, директора и заместителя директора (с одинаковыми окладами для одинаковых должностей). Надбавки установите по критериям, сформулированным в п. 10.3, налоги и выплаты — путем копирования формул.

12.3. Применяя автофильтр и функцию ПРОМЕЖУТОЧНЫЕ ИТОГИ, которая вызывается кнопкой *S* на панели инструментов *Стандартная*, определите:

- сколько в фирме женщин и каков их средний заработок;
- какой максимальный оклад имеет сотрудница, не получающая надбавку;
- кто из мужчин живет в Центральном районе (телефоны начинаются 310...315);

- каков суммарный заработок у менеджеров и инженеров;
- сколько в фирме работает Ивановых и каков их суммарный оклад;
- сколько сотрудников получают больше 500 р., но меньше 1 тыс. р.; сколько из них женщин;
- сколько сотрудников получают больше 900 р. или меньше 500 р. и кто из них не получает надбавки;
- трех самых молодых инженеров;
- а также составьте список Совета Старейшин (трое самых ве-
ликовозрастных, независимо от пола).

13. При поступлении в университет студентам предстоит сдать два экзамена (по математике и физике). Если хоть один экзамен сдан на оценку 3 и ниже, то вероятность поступления равна нулю, если оба экзамена сданы на отлично, то вероятность поступления равна 1, в противном случае — 0,5. Составьте таблицу, позволяющую рассчитать шансы поступления в университет и заполните ее для пяти поступающих с разными результатами вступительных экзаменов.

5.5. ТЕХНОЛОГИЯ ХРАНЕНИЯ, ПОИСКА И СОРТИРОВКИ ИНФОРМАЦИИ

Каждый человек в жизни многократно сталкивается с «базами данных». Это — многочисленные справочники, энциклопедии и т.п.

Базы данных представляют собой информационные модели (см. гл. 6), содержащие данные об объектах и их свойствах. Базы данных хранят информацию о группах объектов с одинаковым набором свойств.

Например, база данных «Записная книжка» хранит информацию о людях, каждый из которых имеет фамилию, имя, телефон и т.д.; библиотечный каталог хранит информацию о книгах, каждая из которых имеет название, автора, год издания и т.д.

Информация в базах данных хранится в упорядоченном виде. Так, в записной книжке все записи упорядочены по алфавиту, а в библиотечном каталоге — либо по алфавиту (алфавитный каталог), либо по области знания (предметный каталог).

База данных (БД) — это информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств.

5.5.1. Типы баз данных

Существует несколько типов баз данных: иерархические, сетевые и реляционные (табличные).

Иерархические базы данных. Иерархические базы данных графически могут быть представлены как перевернутое дерево, со-

стоящее из объектов различных уровней. Первый (верхний) уровень занимает всегда один объект, второй уровень — объекты второго уровня и т. д.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении «предка» (объект более высокого уровня) к «потомку» (объект более низкого уровня); при этом «объект-предок» может не иметь «потомков» или иметь их несколько, тогда как «объект-потомок» обязательно имеет только одного «предка». Объекты, имеющие общего «предка», называются *близнецами* (рис. 5.23, а).

Иерархической базой данных является структура папок Windows, с которой можно работать, запустив «Проводник» (рис. 5.23, б). Первый верхний уровень занимает папка «Рабочий стол». На втором уровне находятся папки «Мой компьютер», «Мои документы», «Сетевое окружение» и «Корзина», которые являются потомками папки «Рабочий стол», а между собой являются близнецами. В свою очередь, папка «Мой компьютер» является предком по отношению к папкам третьего уровня, папкам дисков (Диск 3,5 (A:), (C:), (D:)) и системным папкам («Принтеры», «Панель управления» и др.). Иерархической базой данных является реестр Windows, в котором хранится вся информация, необходимая для нормального функционирования компьютерной системы (данные о конфигурации компьютера и установленных драйверах, об установленных программах, настройки графического интерфейса и др.).

Сетевые базы данных. Сетевая база данных является обобщением иерархической за счет допущения объектов, имеющих более одного предка, т. е. каждый элемент вышестоящего уровня может быть связан одновременно с любыми элементами следующего уровня. На связи между объектами в сетевых моделях не накладывается никаких ограничений (рис. 5.23, в). Сетевой базой данных фактически является «всемирная паутина» глобальной компьютерной сети Интернет. Гиперссылки связывают между собой сотни миллионов документов в единую сетевую базу данных. Примером сетевой базы данных может служить таблица игр на чемпионате по футболу (рис. 5.23, г).

Реляционные (табличные) базы данных. В настоящее время предпочтение отдается реляционным базам данных.

Реляционная база данных содержит перечень объектов одного типа, т. е. объектов, имеющих одинаковый набор свойств. Такую базу данных удобно представлять в виде двумерной таблицы: в каждой ее строке последовательно размещаются значения свойств одного из объектов; каждое значение свойства — в своем столбце, озаглавленном именем свойства, например таблица с данными о студентах.

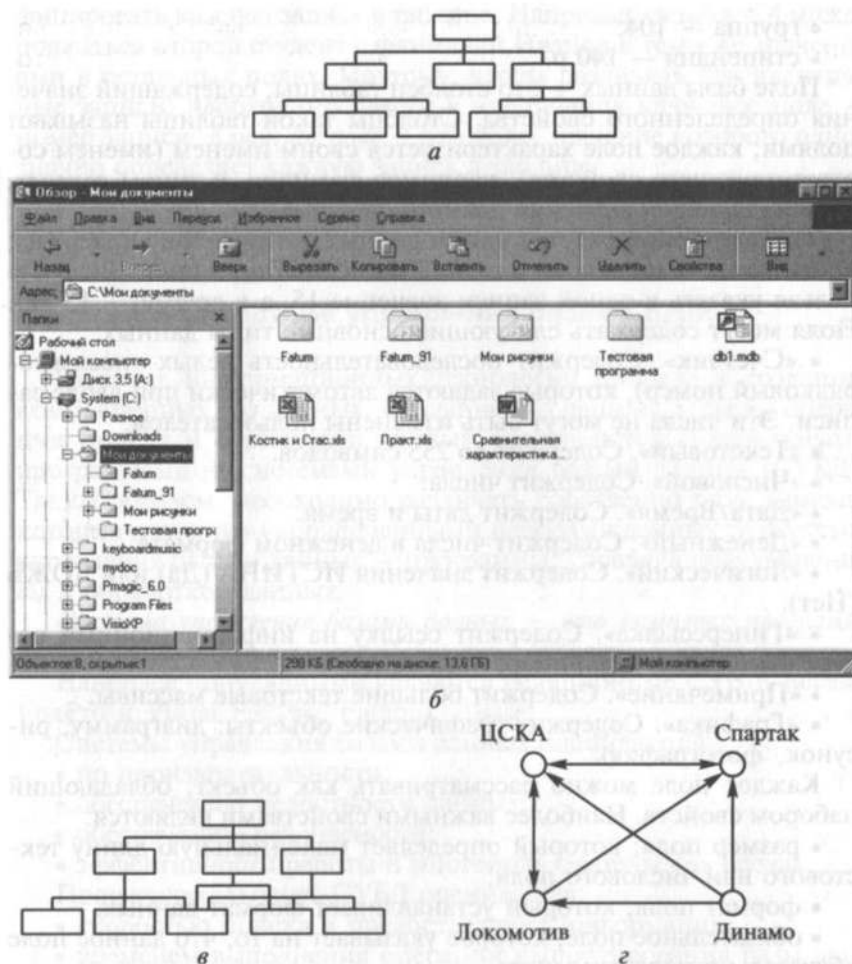


Рис. 5.23. Иерархическая модель и ее реализация на примере структуры папок Windows:

a — иерархическая модель; б — структура папок Windows; в — сетевая модель; z — пример сетевой базы данных

Запись базы данных — это строка таблицы, содержащая набор значений свойств, принадлежащих одному объекту.

Строки таблицы являются записями об объекте; эти записи разбиты на поля столбцами таблицы, поэтому каждая запись представляет набор значений, содержащихся в полях. В табл. 5.4 первая запись несет информацию об объекте 1, обладающем свойствами:

- фамилия — Иванов;
- возраст — 15;

- группа — 104;
- стипендия — 140 р.

Поле базы данных — это столбец таблицы, содержащий значения определенного свойства. Столбцы такой таблицы называют полями; каждое поле характеризуется своим именем (именем соответствующего свойства), например фамилия, и типом данных, представляющих значения данного свойства.

Тип поля определяется типом данных, которые оно содержит; он является одинаковым для всех записей. Так, в поле «Возраст» нельзя указать в одной записи значение 15, а в другой — 15 лет. Поля могут содержать следующие основные типы данных.

- «Счетчик». Содержит последовательность целых чисел (порядковый номер), которые задаются автоматически при вводе записи. Эти числа не могут быть изменены пользователем.
- «Текстовый». Содержит до 255 символов.
- «Числовой». Содержит числа.
- «Дата/Время». Содержит даты и время.
- «Денежный». Содержит числа в денежном формате.
- «Логический». Содержит значения ИСТИНА (Да) или ЛОЖЬ (Нет).
- «Гиперссылка». Содержит ссылку на информационный ресурс Интернета (например, Web-сайт).
- «Примечание». Содержит большие текстовые массивы.
- «Графика». Содержит графические объекты: диаграмму, рисунок, фотографию.

Каждое поле можно рассматривать как объект, обладающий набором свойств. Наиболее важными свойствами являются:

- размер поля, который определяет максимальную длину текстового или числового поля;
- формат поля, который устанавливает формат данных.
- обязательное поле, которое указывает на то, что данное поле обязательно надо заполнить.

Каждая таблица должна содержать, по крайней мере, одно ключевое поле, содержимое которого уникально для каждой записи в этой таблице. Ключевое поле позволяет однозначно иденти-

Таблица 5.4

Данные о студентах

Фамилия	Возраст	Группа	Стипендия
Иванов	15	104	140 р.
Петров	16	108	0 р.
Сидоров	15	101	140 р.
...	—	—	—

фицировать каждую запись в таблице. Например, в табл. 5.4 может появиться второй студент с фамилией Иванов и теми же значениями в остальных полях. Поэтому, чтобы различить две идентичные записи, необходимо ввести в эту таблицу ключевое поле — номер записи. *Ключевое поле* — это поле, значение которого однозначно определяет каждую запись в таблице.

Например, табл. 5.4 нужно добавить поле «№ студ. билета» и оно будет ключевым.

5.5.2. Система управления базами данных

Развитие информационных технологий привело к созданию компьютерных баз данных. Создание баз данных, а также операции поиска и сортировки данных выполняются специальными программами — системами управления базами данных (СУБД). Таким образом, необходимо различать собственно базы данных, которые являются упорядоченными наборами данных, и системы управления базами данных — программы, управляющие хранением и обработкой данных.

Система управления базами данных — это комплекс программ, позволяющий создавать, осуществлять обработку и поиск данных.

Наиболее популярными являются реляционные СУБД: Access, FoxPro и Visual FoxPro, Paradox, dBase.

Системы управления базами данных оцениваются:

- по производительности;
- обеспечению целостности данных;
- обеспечению безопасности;
- эффективности работы в многопользовательских средах.

Производительность СУБД оценивается:

- скоростью поиска и предоставления информации;
- временем выполнения операции импортирования базы данных из других форматов;
- скоростью выполнения таких массовых операций, как обновление, вставка, удаление данных;
- максимальным числом параллельных обращений к данным в многопользовательском режиме.

Обеспечение целостности данных подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной независимо от того, каким образом данные заносятся в память (в интерактивном режиме, посредством импорта или с помощью специальной программы).

Обеспечение безопасности характеризуется наличием средств, обеспечивающих выполнение следующих операций:

- шифрование самих прикладных программ;

- шифрование данных;
- парольная защита;
- ограничение прав доступа.

Работа с БД в любой СУБД включает в себя следующие этапы:

- 1) создание структуры БД;
- 2) ввод и редактирование данных;
- 3) обработка данных в БД.

Создание структуры базы данных. База данных может состоять из одной или нескольких таблиц. Под структурой базы данных будем понимать совокупность таблиц и связи между ними.

Перед созданием структуры БД необходимо располагать описанием выбранной предметной области, которая должна включать в себя описание реальных объектов и процессов, определение всех необходимых источников информации для удовлетворения предполагаемых запросов пользователя и определить потребности в обработке данных, что отражается в информационно-логической модели (ИЛМ).

Рассмотрим формирование структуры базы данных предметной области «Колледж», которая состоит из трех таблиц: «Студенты», «Преподаватели», «Успеваемость».

Изобразим графически структуру БД «Колледж» (рис. 5.24).

Из рис. 5.24 видно, что таблицы можно связать по одинаковым полям.

Например, для связи таблиц «Студенты» и «Преподаватели» можно использовать поле «Группа», а для связи таблиц «Студенты» и «Успеваемость» — поле «№ студ. билета».

Приступим к созданию разработанной структуры в СУБД Access.

Основные приемы работы с базой данных рассмотрим в СУБД Access. СУБД Access работает под управлением Windows. Таким образом, все преимущества Windows доступны в Access (напри-

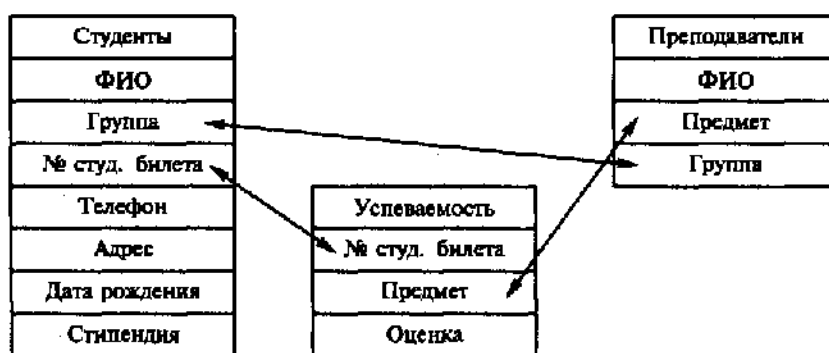


Рис. 5.24. Структура БД «Колледж»

мер, вы можете вырезать, копировать и вставлять данные из любого приложения Windows в приложение Access и наоборот).

Access — это реляционная СУБД. Это означает, что с ее помощью можно работать одновременно с несколькими таблицами базы данных. Таблицу Access можно связать с данными, хранящимися на другом компьютере или на сервере, а также использовать таблицу, созданную в СУБД Paradox или Dbase. Данные Access очень просто комбинировать с данными Excel.

В Access используется созданный для среды Windows Office многооконный интерфейс, но, в отличие от других приложений, не многодокументный. Единовременно может быть открыта одна база данных (так как Access хранит все объекты в одном файле), содержащая обязательное окно базы данных и окна для работы с объектами. В каждый момент времени одно из окон является активным и в нем курсором отмечается активный объект.

После запуска Microsoft Access одновременно с его окном выводится первое диалоговое окно (рис. 5.25, а), позволяющее начать создание БД или открыть существующую БД.

В окне БД в разделе «Объекты» представлены в виде вертикального ряда кнопок все объекты, которые могут быть созданы в БД: «Таблицы», «Запросы», «Формы», «Отчеты», «Макросы» и «Модули» (рис. 5.25, б). При создании новой базы данных список для любого выбранного типа объекта пуст.

Таблицы. В базах данных вся информация хранится в двумерных таблицах. Это базовый объект БД, все остальные объекты создаются на основе существующих таблиц (производные объекты).

Запросы. В СУБД запросы являются важнейшим инструментом. Главное предназначение запросов — это отбор данных на основании заданных условий. С помощью запроса из базы данных можно выбрать информацию, удовлетворяющую определенным условиям.

Формы. Формы позволяют отображать данные, содержащиеся в таблицах или запросах в более удобном для восприятия виде. При помощи форм можно добавлять в таблицы новые данные, а также редактировать или удалять существующие. Форма может содержать рисунки, графики и другие внедренные объекты.

Отчеты. Отчеты предназначены для печати данных, содержащихся в таблицах и запросах, с использованием элементов оформления.

Макросы. Макросы служат для автоматизации повторяющихся операций. Запись макроса производится также, как в других приложениях, например в приложении Word.

Модули. Они также служат для автоматизации работы с БД. Модули еще называют процедурами обработки событий, они пишутся на языке VBA (см. гл. 8).

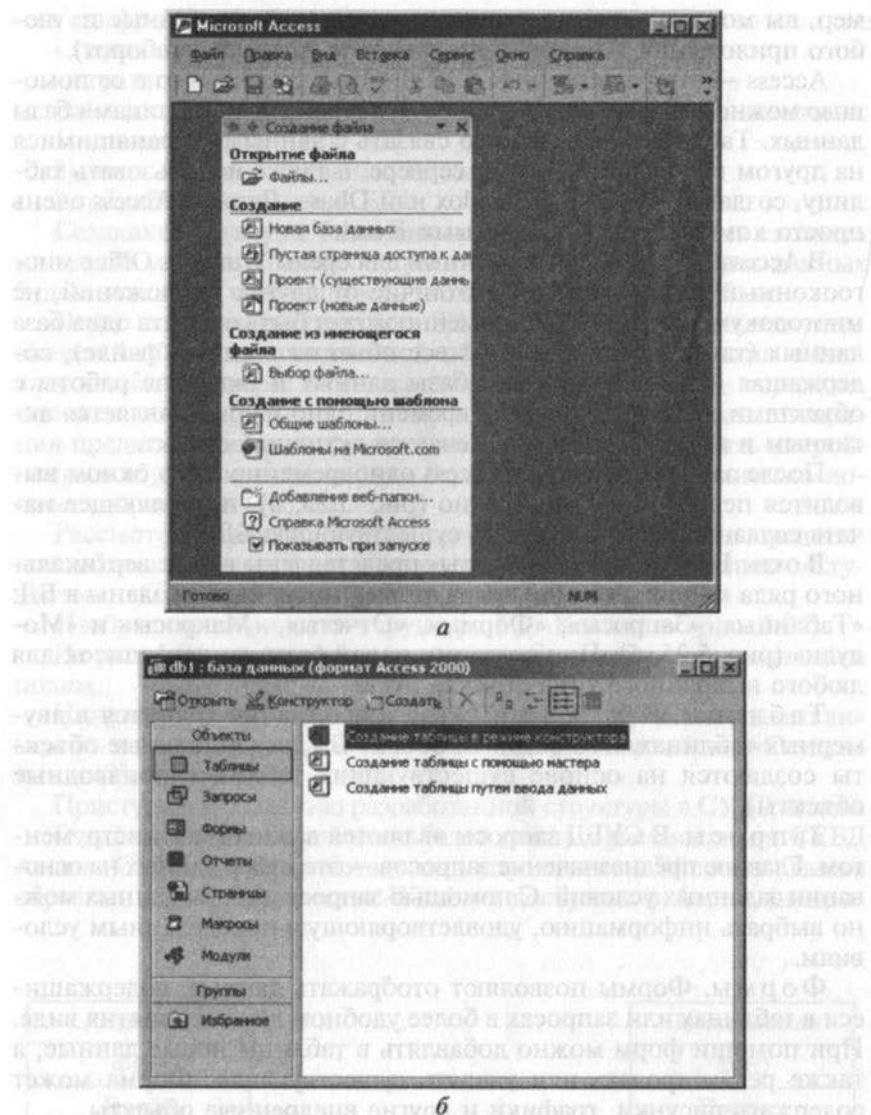


Рис. 5.25. Окно приложения Access:

a — первое диалоговое окно Access; *b* — окно пустой базы данных

Создание БД начинается с создания таблиц. В Access существует четыре способа создания таблиц:

1) использование мастера баз данных для создания всей базы данных, содержащей все требуемые отчеты, таблицы и формы, за одну операцию. Мастер баз данных создает новую базу данных,

его нельзя использовать для добавления новых таблиц, форм, отчетов в уже существующую базу данных;

2) мастер таблиц позволяет выбрать поля для данной таблицы из множества определенных ранее таблиц, таких как «Деловые контакты», «Список личного имущества» или «Рецепты»;

3) ввод данных непосредственно в пустую таблицу в режиме таблицы. При сохранении новой таблицы в Access данные анализируются и каждому полю присваивается необходимый тип данных и формат;

4) определение всех параметров макета таблицы в режиме конструктора.

Независимо от метода, примененного для создания таблицы, всегда имеется возможность использовать режим конструктора для дальнейшего изменения макета таблицы, например для добавления новых полей, установки значений по умолчанию или создания масок ввода. Однако только четвертый метод позволяет сразу задать ту структуру таблицы, которая вам нужна, поэтому рассмотрим именно этот метод.

Создание таблицы БД состоит из двух этапов. На первом этапе определяется ее структура: состав полей, их имена, тип каждого поля, размер поля, ключи, индексы таблиц (рис. 5.26, а). На втором этапе производится заполнение таблицы записями (рис. 5.26, б).

Ввод данных в ячейки таблицы производится обычным образом — вы переводите курсор в нижнюю ячейку, т.е. делаете ее активной, и вводите в нее данные (вводить данные в ячейки с типом данных «Счетчик» не требуется, так как значения в этом поле появляются автоматически при вводе данных в любую другую ячейку данной записи). Особенности ввода следующие: при нажатии клавиши [Del] ячейка очищается; если ввод данных в ячейку прервать, нажав клавишу [Esc], то восстанавливается старое значение, а если нажать клавишу [Enter] или [Tab], то в ячейку заносится новое значение. Можно редактировать текущее значение, меняя в нем отдельные символы. Для этого надо в ячейке таблицы дважды щелкнуть мышью или нажать клавишу [F2]. Однако есть одна особенность при вводе данных — для некоторых типов данных («Числовой», «Денежный», «Дата/время», «Логический») Access автоматически проверяет правильность их ввода. Например, если вы введете букву в ячейку с числовым типом, то Access выдаст сообщение о неправильно введенном значении и не позволит перейти к другой ячейке, пока не будут введены правильные данные.

Для всех типов полей (кроме типов «Счетчик» и поле объекта «OLE») можно самостоятельно задавать ограничения для вводимых данных. Для этого в режиме *Конструктора* надо выбрать вкладку *Общие*, перевести курсор в поле «Условия» на значение и ввести

Таблица1 : таблица

Имя поля	Тип данных
ФИО	Текстовый
Группа	Текстовый
№ студ билета	Числовой
Телефон	Текстовый
Адрес	Текстовый
Дата рождения	Дата/время
Стипендия	Числовой

Сохранение

Имя таблицы: Студенты

OK Отмена

Общие Подстановка

Формат поля Да/Нет

Подпись

Значение по умолчанию

Условие на значение

Сообщение об ошибке

Обязательное поле Нет

Индексированное поле Нет

а

Студенты : таблица

	Код	ФИО	Группа	№ студ билета	Телефон	Адрес	Дата рождения	Стипендия
▶		Сидорова Е.П.	104		1211 120-00-09	Ул. Сиреневая, 3,1	02.07.1988	■
	4	Новиков С.Л.	104		1012 389-87-60	Ул. Денежная, 65,1	08.08.1988	■
*	(Счетчик)			0				■

б

Рис. 5.26. Создание таблиц:

а — окно конструктора таблиц; б — заполненная таблица

ограничения на данные. Например, стаж работы вряд ли может быть больше 70 лет, поэтому ограничения будут следующие: <70. Ограничение можно вводить и на текстовое поле. Обычно в этом случае задаются слова, которые могут присутствовать в данном поле.

Можно использовать еще один удобный инструмент при вводе данных — параметр *Значение по умолчанию* (находится также на вкладке *Общие*).

Здесь можно задать данные, которые Access будет вводить по умолчанию при заполнении таблицы. Это удобно использовать, когда большинство значений данного поля одинаковы и лишь некоторые отличаются. Например, у большинства студентов есть страховые полисы. Если в строку параметра *Значение по умолчанию* ввести значение «Да», то все ячейки поля «Полис» примут значение «Да» и надо будет лишь изменить значения ячеек для студентов, не имеющих полис. Данные можно вводить и копированием их из одной ячейки в другую стандартными средствами Windows.

Редактирование данных производится очень просто — курсор переводится в нужную ячейку, старые данные удаляются (клавишей [Del] или [BackSpace]) и вводятся новые данные. Удалять можно не только данные в ячейках, но и целиком строки, предварительно их выделив. Но если таблица большая, то редактируемые данные надо сначала найти. Это делается следующим образом: переводят курсор в любую ячейку поля, начиная с которой будет вестись поиск, затем открывают пункт меню *Правка* и выполняют команду *Найти*. В появившемся окне вводят образец искомых данных и щелкают по кнопке *Найти*. Если значение найдено, то курсор перейдет в эту ячейку. Иногда требуется большое число одинаковых данных заменить на другое значение. Для этого надо открыть пункт меню *Правка* и выполнить команду *Заменить*. В появившемся окне ввести образцы того, что надо найти и на что заменить. Затем надо решить, будете ли вы просматривать каждое меняемое значение или нет. Если будете, то надо щелкнуть по кнопке *Заменить*, а если нет — один раз щелкнуть по кнопке *Заменить все*.

Ввод и редактирование данных. Записи таблиц БД можно просматривать и редактировать в режиме *Таблицы* или *Формы*. Режим *Таблица* позволяет наблюдать несколько записей одновременно, и в этом состоит достоинство табличного представления.

Однако часто режим *Таблица* не позволяет видеть полностью всю информацию на экране. Если таблица содержит достаточно много полей, а значения полей содержат много символов, то не все поля таблицы могут помещаться на экране, а значения полей могут быть видны не полностью.

Форма одновременно отображает одну запись в удобном для пользователя виде. В процессе создания формы нужно указать, какие поля таблицы включить в форму, как расположить поля в окне формы, а также как можно сделать форму визуально привлекательной.

Фактически с помощью *Формы* создается графический интерфейс доступа к таблице, который может содержать различные управляющие элементы (текстовые поля, кнопки, переключатели и т. д.), а также надписи. Обычно на *Форме* размещаются надписи, являющиеся именами полей таблицы, и текстовые поля, содержащие данные. Пользователь может изменять дизайн формы (размер, цвет и т. д.) управляющих элементов и надписей.

Создать форму можно несколькими способами. Если вы в режиме базы данных откроете вкладку *Формы* и щелкнете по кнопке *Создать*, то откроется окно, в котором указаны способы создания формы.

Конструктор позволит вам создать форму самостоятельно, но это для начинающих пользователей довольно сложно. *Мастер форм* дает возможность автоматически создать форму на основе выб-

ранных полей. Этот режим наиболее удобен при создании форм как для начинающих, так и для опытных пользователей. Access в режиме диалога выясняет у пользователя, какую форму он хочет получить, и создает ее автоматически. Если вас в этой форме что-либо не удовлетворяет, то вы можете исправить это в режиме *Конструктор*. *Автоформы* являются частным случаем мастера форм, т. е. они автоматически создают заданные виды форм практически без участия пользователя. Это может быть удобно, когда базовая таблица одна, содержит не много полей и нужно быстро создать простую форму. Диаграмма создает форму со встроенной диаграммой, а сводная таблица — со сводной таблицей Excel. Эти режимы предпочтительнее использовать опытным пользователям, хотя форму с диаграммой не так сложно создать и начинающему пользователю — Access в этом поможет.

Создадим форму для работы с таблицей «Студенты» с помощью *Мастера форм*. В этом режиме вы можете выбрать поля таблицы для отображения в форме, стиль и цвет оформления фона и ячеек, а также вид формы. *Мастер форм* предлагает четыре вида форм представления данных: в один столбец (рис. 5.27, а), ленточную (рис. 5.27, б), табличную (рис. 5.27, в) и выровненную (рис. 5.27, г).

Настраивать внешний вид и возможности ввода, обработки и просмотра данных можно в режиме *Конструктор*. Вы можете использовать огромное число *Инструментов* и *Свойств формы*, но, чтобы реализовать эти возможности, надо обладать определенным опытом работы.

Перемещение в форме почти идентично перемещению в строках и столбцах. Перемещаться по форме можно, просто щелкая по нужному полю и внося изменения или дополнения в данные.

После открытия форма содержит запись №1. При работе с формой для перехода от одной записи к другой необходимо воспользоваться панелью *Запись*, которая находится в нижней части окна формы. Запись содержит кнопки со стрелками, щелчки по которым позволяют перемещаться по записям, а также поле номера записи, позволяющее ввести номер искомой записи.

Формы необходимы для удобства ввода, просмотра и редактирования данных.

Обработка данных в БД. Обработка данных (поиск, сортировка в базы данных и запросы) производится с помощью средств, встроенных в систему управления базами данных (СУБД).

Быстрый поиск данных. Access позволяет производить поиск записей, в которых значения определенного поля полностью или частично совпадают с некоторой величиной.

Например, в БД «Колледж» мы хотим найти запись, содержащую фамилию Новихин. Можно ввести лишь часть значения поля и осуществить поиск записи.

Для этого необходимо:

- 1) открыть таблицу «Студенты» БД «Колледж», дважды щелкнув по соответствующему значку в окне БД;
- 2) ввести команду *Правка/Найти*. Появится диалоговая панель *Поиск* (рис. 5.28, а). В поле «Образец» необходимо ввести искомый

Студенты

Код

ФИО: Иванов А.В.

Группа: 101

№ студ билета: 1001

Телефон: 234-67-89

Адрес: Ул. Перерва, 7, 54

Дата рождения: 12.03.1987

Стипендия: ☒

Запись: 1 из 3

а

Студенты

Код	ФИО	Группа	билета	Телефон	Адрес	ждения	Сти
1	Иванов А.В.	101	1001	234-67-89	Ул. Перерва, 7, 54	03.1987	☒
2	Петров С.А.	102	1200	433-98-00	Ул. Новая, 67, 2	09.1988	☐
3	Сидорова Е.П.	104	1211	120-00-09	Ул. Сиреневая, 3, 1		☐
*	(Счетчик)		0				☐

б

Студенты

Код	ФИО	Группа	№ студ б	Телефон	Адрес	Дата рож
1	Иванов А.В.	101	1001	234-67-89	Ул. Перерва, 7, 54	12.03.1987
2	Петров С.А.	102	1200	433-98-00	Ул. Новая, 67, 2	29.09.1988
3	Сидорова Е.П.	104	1211	120-00-09	Ул. Сиреневая, 3, 1	
*	(Счетчик)		0			

в

Студенты

Код	ФИО	Группа	№ студ билета	Телефон	Адрес	Дата рождения
1	Иванов А.В.	101	1001	234-67-89	Ул. Перерва, 7, 54	12.03.1987
2	Петров С.А.	102	1200	433-98-00	Ул. Новая, 67, 2	29.09.1988
3	Сидорова Е.П.	104	1211	120-00-09	Ул. Сиреневая, 3, 1	
*	(Счетчик)		0			

Стипендия: ☒

Запись: 1 из 3

г

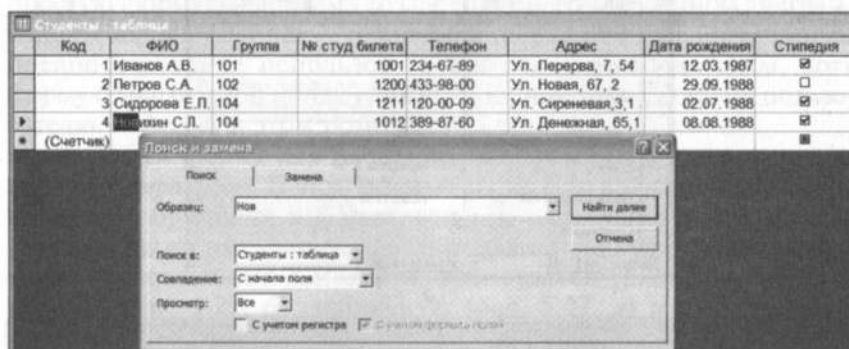
Рис. 5.27. Образцы форм:

а — в один столбец; б — ленточная; в — табличная; г — выровненная

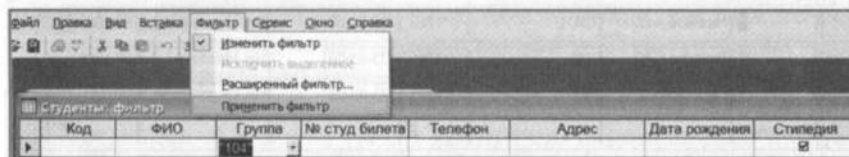
текст, а в поле «Совпадение» — выбрать пункт с любой частью поля.

В результате будет найдена и отмечена как активная запись № 4 (рис. 5.28, б).

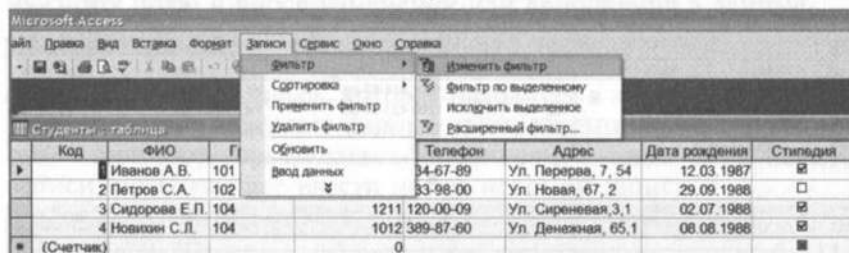
Поиск данных с помощью фильтров. Гораздо больше возможностей для поиска данных в БД предоставляют фильтры. *Фильтр* — это набор условий, применяемых для отбора подмножества запи-



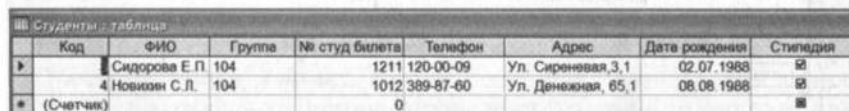
а



б



в



г

Рис. 5.28. Обработка данных:

а — поиск и замена; б — простой фильтр; в — результаты поиска по фамилии Иванов; г — результаты отбора по полям «Группа» и «Стипендия»

сей. Условия отбора записей создаются с использованием операторов сравнения (=, >, < и т.д.).

Простые фильтры содержат условия отбора записей только для одного поля. Сложные фильтры содержат несколько условий для различных полей. В результате применения сложного фильтра будут отобраны только те записи, которые удовлетворяют всем условиям одновременно.

Фильтрация данных в Access производится с помощью кнопок *Фильтр* по выделенному или *Изменить фильтр* (команда меню *Записи, Фильтр, Изменить фильтр*). После нажатия второй кнопки от таблицы остается одна запись. Каждое поле становится полем со списком (когда в нем находится курсор), в котором можно выбрать из списка все значения для данного поля. После щелчка по кнопке *Применить фильтр* будут выбраны записи, соответствующие измененному фильтру. Еще более сложные условия фильтрации можно задать командой меню *Записи, Фильтр, Расширенный фильтр*.

Создадим простой фильтр для таблицы «Студенты», который отбирает записи по следующим условиям: группа — 104, стипендия — выплачивается.

1. Открыть таблицу «Студенты», дважды щелкнув по соответствующему значку в окне БД.

2. Ввести команду *Записи/Фильтр/Изменить фильтр*. В появившемся окне (рис. 5.28, б) ввести условия поиска в соответствующих полях. Фильтр создан.

3. Ввести команду *Записи/Применить фильтр*. В появившемся окне таблицы будут выведены записи, удовлетворяющие условиям поиска. В данном случае найдено трое студентов (рис. 5.28, г).

Сортировка данных. Базы данных могут содержать сотни и тысячи записей. Часто бывает необходимо их упорядочить, т.е. расположить в определенной последовательности. Упорядочение записей называется сортировкой.

Сортировка записей производится по какому-либо полю. Значения, содержащиеся в этом поле, располагаются в определенном порядке, который определяется типом поля:

- по алфавиту, если поле текстовое;
- по величине числа, если поле числовое;
- по дате, если тип поля «Дата/Время» и т.д.

Сортировка данных может производиться либо по возрастанию, либо по убыванию их значений. В процессе сортировки целостность записей сохраняется, т.е. они переносятся из одного места таблицы в другое целиком.

Произведем сортировку таблицы «Студенты», например по полю «Дата рождения» (рис. 5.29).

Быстрая сортировка данных. Она осуществляется следующим образом.

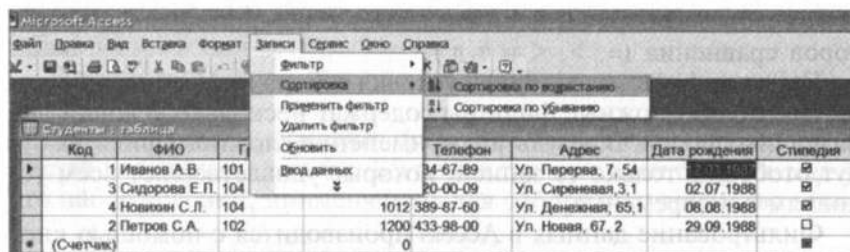


Рис. 5.29. Сортировка

1) В окне *Колледж: база данных* в группе объектов «Таблицы» выделить таблицу «Студенты» и щелкнуть по кнопке *Открыть*.

2) Выделить поле фамилия и вести команду *Записи/Сортировка/Сортировка по возрастанию*. Записи в таблице будут отсортированы.

Могут реализовываться вложенные сортировки, т.е. сортировки, которые последовательно производятся по нескольким полям. После сортировки по первому столбцу производится сортировка по второму столбцу и т.д.

Поиск данных с помощью запросов. Запросы осуществляют поиск в данных БД так же, как и фильтры. Различие между ними заключается в том, что запросы являются самостоятельными объектами в БД, а фильтры привязаны к конкретной таблице. Запрос является производным объектом от таблицы. Однако результатом выполнения запроса является также таблица, т.е. запросы могут использоваться вместо таблиц. Например, форма может быть создана как для таблицы, так и для запроса.

Запросы позволяют отобрать те записи, которые удовлетворяют заданным условиям. Запросы, как и фильтры, бывают простые и сложные. Простой запрос содержит одно условие, а сложный запрос содержит несколько условий для различных полей. В процессе создания запроса можно отбирать не только записи, но и поля, которые будут присутствовать в запросе.

Создадим сложный запрос.

1. В окне *Колледж: база данных* выделить группу «Запросы» и выбрать пункт «Создание запроса с помощью конструктора».

2. На диалоговой панели *Добавление таблицы* выбрать таблицу «Студенты», для которой создается запрос. Щелкнуть по кнопке *Добавить*.

3. В окне запроса в строке «Поле» из раскрывающегося списка выбрать имена полей, в которых будут заданы условия. В строке «Условие отбора» ввести условия для выбранных полей (рис. 5.30, а).

4. В строке «Вывод на экран» задать поля, которые будут представлены в запросе.

5. Сохранить запрос под именем «Запрос1» с помощью команды *Файл/Сохранить как*.

6. В окне *Колледж: База данных* выделить «Запрос1» и щелкнуть по кнопке *Открыть*. В появившемся окне запроса будут выведены записи, удовлетворяющие условиям поиска. В данном случае найдено три студента.

Access позволяет выполнять вложенные сортировки с помощью запросов.

В окне *Колледж: база данных* выделить группу «Запросы» и выбрать пункт «Создание запроса с помощью конструктора».

На диалоговой панели *Добавление таблицы* выбрать таблицу «Студенты», для которой создается запрос. Щелкнуть по кнопке *Добавить*.

В окне запроса в строке «Сортировка» из раскрывающегося списка выбрать имена полей, в которых будет задан тип сортировки. В строке «Вывод на экран» задать поля, которые будут представлены в запросе.

Сохранить запрос под именем «Запрос2» с помощью команды *Файл/Сохранить как*. На вкладке *Запросы* выделить «Запрос2» и щелкнуть по кнопке *Открыть*. В появившемся окне запроса будет выведена отсортированная таблица.

Запрос с параметром. В частном случае, проектируя запрос, можно создать из него небольшую программу, которая будет работать по-разному, в зависимости от вводимого *параметра*. Идею такого запроса рассмотрим на простейшем примере.

а

Код студента	Наличие стипендии	Номер группы	Фамилия	Имя	Отчество
3	☐	151	Кривинский	Сергей	Николаевич
7	☐	151	Перлов	Кирилл	Николаевич
8	☐	151	Соколова	Наталья	Петровна

б

Рис. 5.30. Запрос на выборку:

а — условия; б — результат

Допустим, вы создали запрос-выборку для БД «Колледж» с двумя столбцами: «Фамилия» и «Стипендия».

Далее вы хотите создать программу, которая выдает на экран не весь список студентов (он может быть очень длинным), а только студентов избранной вами группы. Чтобы создать такую программу, выполните следующие шаги:

- 1) откройте запрос в режиме *Конструктор*;
- 2) в качестве третьего столбца вставьте столбец «Группа» (рис. 5.31, а);
- 3) погасите флажок выдачи этого столбца на экран, а в поле «Условие отбора» (в этом столбце) наберите текст приглашения: [Введите номер группы] (квадратные скобки обязательны);
- 4) сохраните запрос (если хотите, под другим именем).

Теперь, когда вы откроете этот запрос в окне базы данных, на экране появится диалоговое окно с приглашением «Введите номер группы» (рис. 5.31, б). В ответ на это приглашение вы набираете группу (например, 104) и нажимаете кнопку *ОК*. На экран будет выведен список (рис. 5.31, в).

Пользуясь этой технологией, вы можете создавать более сложные запросы с параметрами.

Отбор данных в БД может производиться с помощью установки фильтра механизма поиска или форматирования запроса.

Отчеты. Отчет выполняет в программе Microsoft Access особую «презентативную» миссию, обеспечивая вывод на экран или бумажный носитель информации из базы данных в наиболее удобном для восприятия и работы виде.

Если другие объекты, например таблицы и формы, используются для ввода данных, запросы позволяют редактировать данные и выполнять поиск необходимых записей, то благодаря отчетам все важные сведения, хранимые в таблицах, могут быть переданы другим людям для дальнейшего использования, причем как в безбумажном (электронном) формате, так и в виде печатных документов.

Отчеты могут содержать данные из одной или нескольких таблиц, либо запросов и быть как одностраничными (например, сводная ведомость об успеваемости по каждой группе колледжа), так и многостраничными. В отчеты можно поместить графики и рисунки, включить нумерацию страниц и выводить значения вычисляемых полей. Записи, отображаемые в отчете, можно отсортировать по одному или нескольким полям, а также сгруппировать, применяя к таким группам необходимые вычисления.

Основные типы отчетов, которые можно создать средствами Microsoft Access, приведены ниже.

В столбец. В таком отчете, напоминающем макет формы, все поля выводятся на экран как один столбец. Слева от каждого поля отображается значение, указанное в свойстве *Подпись данного поля*.

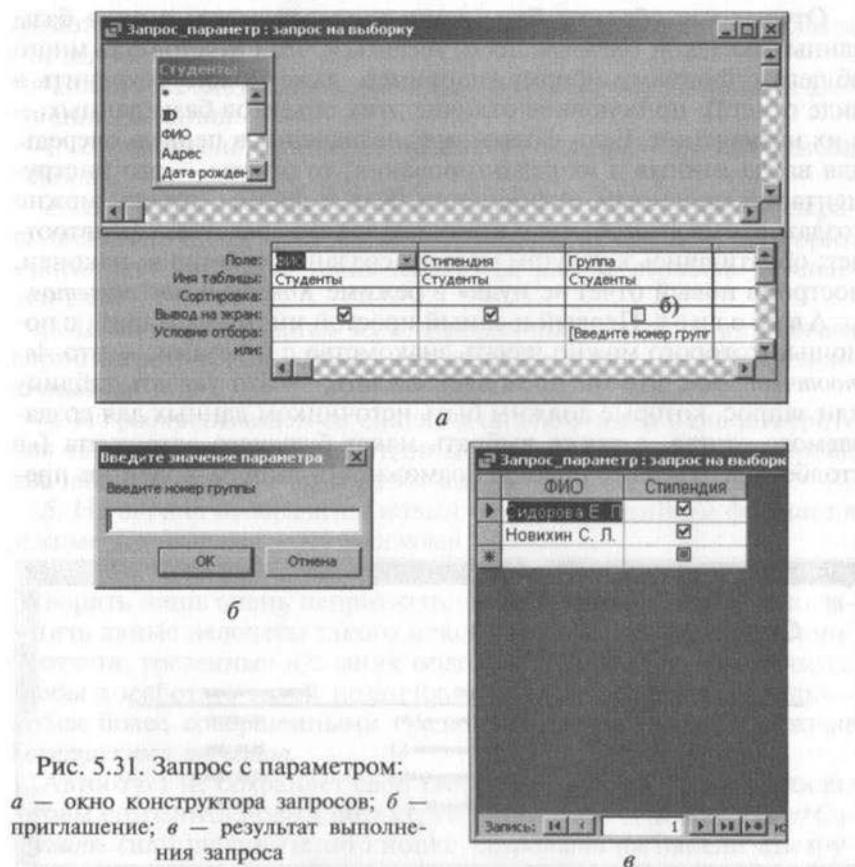


Рис. 5.31. Запрос с параметром:
 а — окно конструктора запросов; б — приглашение; в — результат выполнения запроса

Поля могут быть упорядочены в любом порядке, а на одной странице отчета размещается одна или несколько записей.

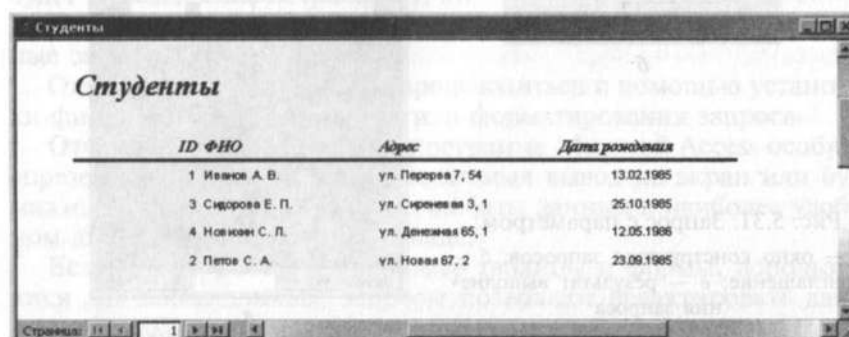
Ленточный. Каждая запись выводится в таком отчете в отдельной строке, а каждое поле — в отдельном столбце. Подписи полей печатаются в качестве заголовков столбцов. Ленточные отчеты позволяют группировать данные по одному или нескольким полям, а также печатать итоговую информацию по группам и по отчету в целом (рис. 5.32, а).

Почтовые наклейки. С помощью мастера можно создать отчет, отформатированный для печати почтовых наклеек. Каждая такая наклейка — это набор данных, расположенных в нескольких полях одной записи (5.32, б).

Отчеты с графиками и диаграммами. В отчет Microsoft Access можно вставить график или диаграмму, разработанную с использованием программы Microsoft Graph.

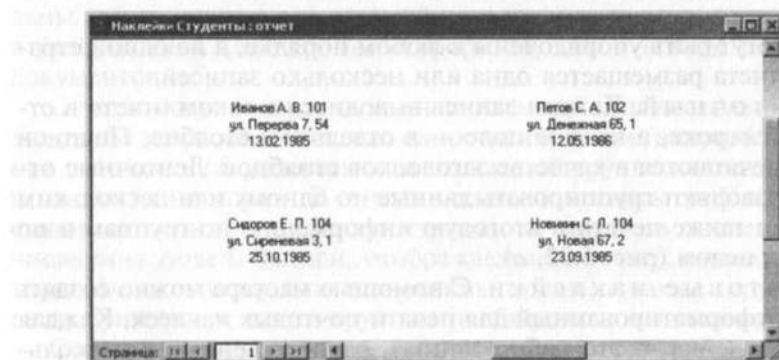
Отчеты как объекты базы данных представлены в окне базы данных вкладкой *Отчеты*. Легко убедиться, что у отчетов есть много общего с формами (форму, например, даже можно сохранить в виде отчета), но основное отличие этих объектов базы данных — в их назначении. Если формы предназначены, в первую очередь, для ввода данных и их редактирования, то отчеты — это инструмент для просмотра информации. Как и формы, отчеты можно создать тремя способами: с помощью такого средства, как автоотчет; обратившись к услугам мастера создания отчетов и, наконец, построив новый отчет «с нуля» в режиме *Конструктор отчетов*.

Автоотчет. Первый и самый простой инструмент, с помощью которого можно начать знакомство с отчетами, — это *Автоотчет*. Все, что он позволяет сделать, — это указать таблицу или запрос, которые должны быть источником данных для создаваемого отчета, а также выбрать макет будущего автоотчета («в столбец» или «ленточный»). Возможность выбора полей не пре-



ID	ФИО	Адрес	Дата рождения
1	Иванов А. В.	ул. Перерва 7, 54	13.02.1985
3	Сидорова Е. П.	ул. Сиреневая 3, 1	25.10.1985
4	Новиков С. П.	ул. Денежная 65, 1	12.05.1986
2	Петов С. А.	ул. Новая 67, 2	23.09.1985

а



Иванов А. В. 101 ул. Перерва 7, 54 13.02.1985	Петов С. А. 102 ул. Денежная 65, 1 12.05.1986
Сидорова Е. П. 104 ул. Сиреневая 3, 1 25.10.1985	Новиков С. П. 104 ул. Новая 67, 2 23.09.1985

б

Рис. 5.32. Отчеты:

а — ленточный; б — для печати готовых наклеек

дусмотрена: в автоотчет помещаются все поля исходной таблицы или запроса.

Чтобы воспользоваться помощью *Автоотчета*, выполните следующие действия.

1. В окне базы данных щелкните по кнопке *Отчеты* панели *Объекты*.

2. Щелкните по кнопке *Создать* панели инструментов. На экране появится диалоговое окно *Новый отчет*, где для категории «Автоотчет» доступны такие варианты, как «в столбец» и «ленточный».

3. Щелкните на одном из предложенных в окне *Новый отчет* форматов автоотчета («Автоотчет: в столбец» или «Автоотчет: ленточный»).

4. Из раскрывающегося списка в нижней части окна выберите имя таблицы (или запроса), данные для которых станут основой для таблицы или отчета, а затем щелкните по кнопке *ОК*.

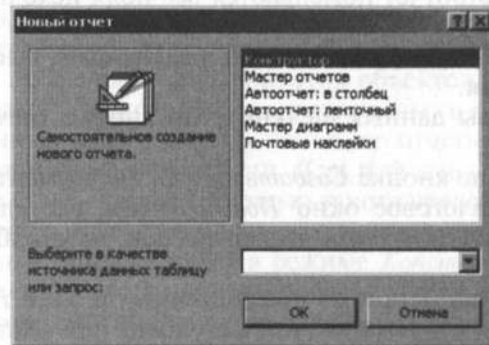
5. На экране отобразится новый отчет в выбранном формате в режиме *Предварительный просмотр*.

Сразу же становится понятно, что подобный отчет может удовлетворить лишь очень непритязательного пользователя. Легко заметить явные недочеты такого макета: неполный заголовок самого отчета, урезанные названия полей и непосредственно данных. Чтобы доработать такой незатейливый отчет, следует воспользоваться более совершенными средствами, доступными в режиме *Конструктор отчетов*.

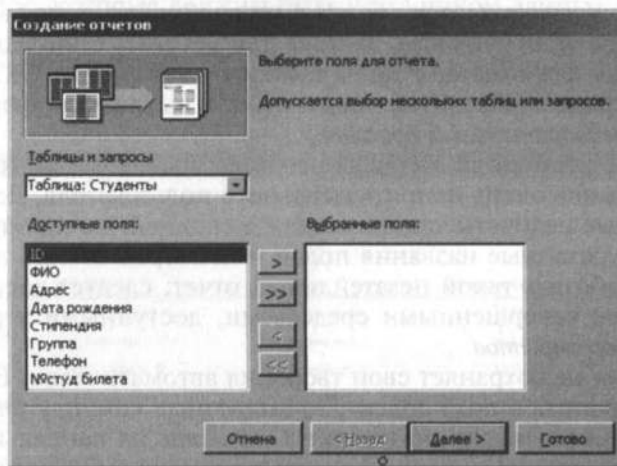
Автоотчет не сохраняет свои творения автоматически. Если вы готовы сохранить новый объект, то выполните команду *Файл/Сохранить* (или щелкните по кнопке *Сохранить* на панели инструментов), укажите в появившемся диалоговом окне *Сохранение: имя отчета* и щелкните по кнопке *ОК*.

Для таблицы, открытой в режиме таблицы, можно быстро создать автоотчет «в столбец». Для этого щелкните по кнопке со стрелкой раскрывающегося списка «Новый объект» панели инструментов *Таблица* в режиме *Таблицы* и выберите пункт «Автоотчет». Если таблица не открыта, а лишь выделена в окне *Базы данных* (вкладка *Таблицы*), то для создания автоотчета «в столбец» выполните команду главного меню *Вставка/Автоотчет*.

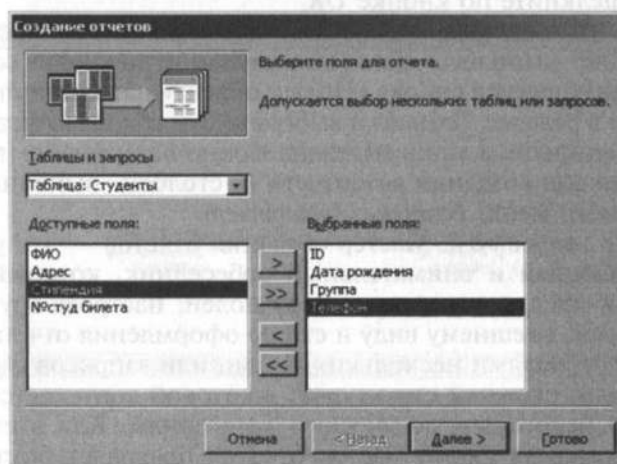
Мастер отчетов. Мастер создания отчетов — это уже более образованный и внимательный собеседник, который готов учесть ваши предложения по выбору полей, настроек группирования записей, внешнему виду и стилю оформления отчета. Мастер работает с полями нескольких таблиц или запросов и создает отчеты с более сложной структурой, в которой допускается группировка по некоторым логическим категориям. Как в случае с мастером создания форм, мастер отчетов позволяет быстро создать структуру отчета, которую можно доработать в режиме *Кон-*



а



б



в

Создание отчетов

Добавить уровень группировки?

ФИО
Адрес
Дата рождения
Стипендия
Группа
Телефон
№ студ билета
предмет
оценка

>
<
+
-

Уровень
+
-

ID
Дата рождения - по месяцам
ФИО, Адрес, Дата рождения, Стипендия, Группа, Телефон, № студ билета, предмет, оценка

Группировка... Отмена < Назад Далее > Готово

г

Создание отчетов

Выберите порядок сортировки и вычисления, выполняемые для записей.

Допускается сортировка записей по возрастанию или по убыванию, включающая до 4 полей.

	1	2	3	4
А	Иванов	Сидорова	Петов	Новиков
Б	Иванов	Сидорова	Петов	Новиков
В	Иванов	Сидорова	Петов	Новиков
Г	Иванов	Сидорова	Петов	Новиков

1. Адрес * по возрастанию
2. * по возрастанию
3. * по возрастанию
4. * по возрастанию
Итого...

Отмена < Назад Далее > Готово

д

Студенты			
ФИО	Адрес	Дата рождения	Стипендия
Иванов А. В.	ул. Перерва 7, 54	13.02.1985	☒
Новиков С. Л.	ул. Денежная 65, 1	12.05.1986	☒
Петов С. А.	ул. Новая 67, 2	23.09.1985	☐
Сидорова Е. П.	ул. Сиреневая 3, 1	25.10.1985	☒

е

Рис. 5.33. Мастер отчетов:

а — диалоговое окно *Новый отчет*; б — выбор полей для отчета; в — результат выбора полей; г — окно группировки данных; д — окно выбора порядка сортировки записей в отчет; е — готовый отчет

структур отчетов. Рассмотрим создание отчета с помощью мастера на примере БД «Колледж».

Чтобы создать отчет под руководством мастера, выполните следующие шаги.

1. Щелкните по кнопке *Отчеты* окна базы данных, а затем дважды щелкните на ярлыке *Создание отчета с помощью мастера*. На экране появится первое окно мастера создания отчетов (рис. 5.33, а).

2. В окне *Новый отчет* выберите из списка способов создания отчетов *Мастер отчетов*.

3. Окно выбора полей. Выберите из раскрывающегося списка «Таблицы и запросы» таблицу (или запрос), которая содержит поля будущего отчета. Для нашего примера первой следует выбрать таблицу «Студенты» (рис. 5.33, б). Список в области «Доступные поля» заполнится полями указанной таблицы или запроса.

4. Включите в отчет подходящее поле из списка «Доступные поля». Чтобы сделать это, щелкните на имени нужного поля, а затем по кнопке со значком «>» (или просто дважды щелкните на имени поля). Выделенное поле переместится в список «Выбранные поля» (рис. 5.33, в). Повторите шаг 3 требуемое число раз, чтобы включить в отчет все необходимые поля.

5. Мастер позволяет включить в новый отчет поля и из других таблиц. Для этого вернитесь к шагу 2 и выберите другую таблицу (между такими таблицами в базе данных должны быть установлены связи). Когда список «Выбранные поля» будет полностью сформирован, щелкните по кнопке *Далее*, чтобы перейти к следующему этапу создания отчета. По умолчанию поля в создаваемом мастеров отчете будут располагаться в той же последовательности, что и в самой таблице. Чтобы изменить порядок отображения полей в отчете, перенесите поля в список «Выделенные поля» в той последовательности, в которой они должны будут располагаться в готовом отчете.

6. Вид представления данных. В следующем окне, которое активизируется только в том случае, если в отчет включены поля из нескольких таблиц, мастер предложит выбрать вид представления данных (рис. 5.33, г). В качестве подсказки справа отображается образец будущей страницы отчета, соответствующий выбранной схеме организации. Наша задача — подготовить отчет, отражающий сведения о студентах. Чтобы вникнуть в процесс группировки, воспользуйтесь подсказкой мастера, щелкнув по кнопке с двумя стрелками *Вывести дополнительные сведения*.

Для задания группировки по определенному полю щелкните на имени требуемого поля в списке слева, а затем — по кнопке со значком «>» (или просто дважды щелкните на имени поля). Чтобы перейти к следующему окну мастера, щелкните по кнопке *Далее*. В отчете можно определить всего четыре уровня группировки. Если

возникла необходимость реорганизовать структуру отчета и выделить некоторые другие логические подразделы отчета, воспользуйтесь кнопками *Уровень*. Кнопка со стрелкой, направленной вверх, повышает уровень выделенного поля в иерархии групп, а кнопка со стрелкой, направленной вниз, — понижает.

7. Сортировка данных и вычисления. Следующее окно мастера позволяет выбрать порядок сортировки для записей отчета, которые будут отображаться непосредственно в группах. Мастер позволяет выбрать до четырех полей сортировки. Чтобы задать поле, по которому будет производиться сортировка, выберите имя нужного поля из раскрывающегося списка под номером 1. Справа от этого списка расположена кнопка, на которой указан порядок сортировки. Кнопка действует как переключатель: чтобы изменить принятый по умолчанию порядок сортировки по возрастанию на обратный (по убыванию), просто щелкните по данной кнопке. Для отчета, создаваемого в рамках нашего примера, полем сортировки является поле «Адрес» (рис. 5.33, *д*). Щелчок по кнопке *Итоги*, расположенной под полями сортировки, откроет одноименное диалоговое окно, в котором можно установить флажки для применения к числовым полям некоторых статических функций, например подсчета общей суммы (Sum), среднего значения (Avg), а также нахождения минимального (Min) или максимального (Max) значения. В отчет будут добавлены поля, соответствующие указанным функциям. Кроме того, щелкнув на нужном переключателе в группе «Показать», можно вывести на печать и данные, и результаты математических вычислений, а можно ограничиться только итоговой информацией, которая будет добавлена в конец каждой группы и всего отчета в целом. Если установлен флажок «Вычислить проценты», то для данного поля программа Access подчитает сумму всех значений и долю в процентах каждого значения в этой сумме. Выполнив нужные настройки, щелкните по кнопке *Далее*.

8. Макет отчета. Новое окно предлагает выбрать макет отчета. Список доступных опций макета представлен справа в виде группы переключателей. Слева расположена область предварительного просмотра, в которой при выборе соответствующего переключателя схематично отображается вид будущего отчета. Ознакомившись с предлагаемыми типами макетов, щелкните на переключателе наиболее подходящего. Кроме того, в текущем окне мастера можно выбрать ориентацию нового отчета (книжная или альбомная), щелкнув на соответствующем переключателе группы «Ориентация», а также настроить ширину полей для размещения на одной странице (флажок данной опции по умолчанию установлен). Если для будущего отчета не заданы никакие настройки группирования записей, то в списке типов макетов будут предложены только элементы «в столбец», «табличный» и «выровненный».

9. Следующее окно мастера предлагает выбрать стилевое оформление для отчета. Список всех доступных стилей представлен в правой части окна. Щелкнув на одном из них, в расположенной слева области предварительного просмотра можно увидеть, как будет выглядеть отчет.

10. На заключительном этапе мастер предлагает указать заголовок для отчета. Введите нужный заголовок в соответствующее текстовое поле диалогового окна или оставьте предложенное мастером имя без изменений. Затем примите предложенную по умолчанию опцию *Просмотреть отчет* и щелкните по кнопке *Готово*. Мастер создает отчет с учетом всех указанных вами настроек, сохраняет его под именем, заданном в последнем окне мастера, и открывает отчет в режиме *Предварительный просмотр* (рис. 5.33, е). Окно *Предварительный просмотр* позволяет увидеть все элементы отчета в том виде, в котором они появятся на листе бумаги. Для перемещения между страницами отчета воспользуйтесь кнопками навигационной панели в левом нижнем углу окна. В режиме предварительного просмотра можно выбрать произвольный масштаб изображения отчета, для чего необходимо щелкнуть по кнопке *Масштаб* (с изображением лупы). Чтобы указать конкретное число страниц отчета, отображаемых в режиме предварительного просмотра, воспользуйтесь кнопками *Одна страница*, *Две страницы* или *Несколько страниц* либо выполните команду *Вид/Число страниц*. Если внешний вид отчета вас вполне устраивает, то можно отправить его сразу на печать, щелкнув по кнопке *Печать* или выполнив команду *Файл/Печать*.

Отчеты необходимы для формирования сводных ведомостей, содержащих разнородные данные с возможностью группировки и выполнения вычислений над данными.

Конструктор отчетов. Работа в этом режиме включает в себя все этапы, рассмотренные в создании отчетов с помощью мастера отчетов, но требует более квалифицированной подготовки.

Связь между таблицами и целостность данных. На примере БД «Колледж» мы теоретически рассмотрели понятие реляционной базы данных как совокупности взаимосвязанных таблиц. Теперь мы рассмотрим, как реализуется установка связей между таблицами на практике.

Между одноименными полями двух таблиц MS Access автоматически устанавливает связь. Например, между таблицами «Студенты» и «Успеваемость» устанавливается связь по полю «№ студ. билета». Что это означает? Это означает, что при формировании запроса к этой паре таблиц Access сможет объединить записи (строки) таблиц, в которых значения поля «№ студ. билета» совпадают.

Кроме того, Access позволяет вручную установить связь между таблицами по разноименным полям, однако этой возможностью

лучше не пользоваться, так как это запутывает и аналитиков, и пользователей.

В общем случае допускается связь по двум, трем и более одноименным полям, но для простоты изложения мы этот случай не рассматриваем.

Итак, если установлена связь между двумя таблицами (автоматически или вручную), данные из обеих таблиц можно объединять. Иногда этого достаточно, однако при создании серьезных баз данных нам придется позаботиться о дополнительных средствах контроля связанных данных, вводимых в разные таблицы. Например, при ведении таблицы «Успеваемость» нельзя допустить случайный ввод в эту таблицу данных о несуществующих студентах, нельзя удалять из таблицы «Студенты» записи о студентах, о которых хранятся данные об успеваемости.

Механизм, который обеспечивает согласованность данных между двумя связанными таблицами, называется поддержкой целостности данных. Чтобы обеспечить целостность данных, при установлении связи между двумя таблицами нужно активизировать переключатель. Если пользователь включил механизм поддержки целостности, то он должен одновременно указать тип связи: «один к одному» или «один ко многим».

Целостность данных означает следующее:

- в связанное поле подчиненной таблицы можно вводить только те значения, которые имеются в связанном поле главной таблицы (например, в таблицу «Успеваемость» нельзя ввести запись с номером студенческого билета, отсутствующим в таблице «Студенты»);

- из главной таблицы нельзя удалить запись, у которой значение связанного поля совпадает хотя бы с одним значением того же поля в подчиненной таблице (например, из таблицы «Студенты» нельзя удалить «№ студ. билета», который еще не удален из таблицы «Успеваемость»).

При попытке нарушить эти запреты MS Access выдает сообщение об ошибке.

Включив механизм поддержки целостности, вы можете (но не обязаны) указать, чтобы при модификации данных система запускала следующие процессы:

- каскадное обновление связанных полей;
- каскадное обновление связанных записей.

Каскадное обновление означает, что изменение связанного поля в главной таблице (например, кода клиента) автоматически будет и в связанных записях подчиненной таблицы. Например, если вы замените в таблице «Студенты» № студ. билета, то код 80 во всех записях таблицы «Успеваемость» изменится на 191.

Каскадное удаление означает, что при удалении записи из главной таблицы из подчиненной таблицы будут удалены все записи,

у которых значение связанного поля совпадает с удаляемым значением.

Установление связей между таблицами рассмотрим на конкретном примере — на нашей базе данных «Колледж».

Выберите команду *Правка/Схема данных*. На экране появится диалоговое окно со списком всех таблиц открытой базы данных (рис. 5.34, а).

Вы можете включить в этот список и запросы (или создать список только из запросов).

Ваша задача — указать системе те таблицы, между которыми вы устанавливаете связи. Выделите таблицу «Преподаватели» и нажмите кнопку *Добавить*, затем то же самое проделайте с таблицами «Успеваемость» и «Студенты». Нажмите кнопку *Закрыть*.

На экране появится окно *Схема данных*.

Это окно содержит все таблицы базы данных, между которыми устанавливаются (или уже установлены) связи. Для установления связи между двумя таблицами можно методом «Drag-and-Drop» переместить имя поля с первичным ключом главной таблицы на одноименное поле подчиненной таблицы.

Прежде всего установите связь между таблицами «Студенты» и «Успеваемость». Удерживая нажатой левую кнопку мыши, переместите № студ. билета из таблицы «Студенты» на № студ. билета в таблице «Успеваемость» и отпустите левую кнопку мыши. На экране появится диалоговое окно *Связи* (рис. 5.34, б). В этом окне установите флажок «Обеспечение целостности данных». Этим вы включите механизм поддержки целостности данных в таблицах «Студенты» и «Успеваемость».

После активизации флажка «Обеспечение целостности данных» становятся доступными радиокнопки *Отношение* и два флажка каскадных операций. В группе «Отношение» надо обязательно выбрать один из типов связи: «один-к-одному» или «один-ко-многим».

Кроме того, вы можете (если хотите) установить любой флажок (или оба) каскадной модификации — обновления или удаления. Нажмите кнопку *Создать*. На экране вновь появится окно *Схема данных* с графическим изображением установленной связи (рис. 5.34, в).

Пометки у концов линии связи означают, что одна запись таблицы может иметь сколько угодно связанных записей. Если хотите удалить связь, то щелкните мышью на линии связи (выделите ее), а затем нажмите клавишу [Del]. Теперь, пользуясь связями, можно создать многотабличную форму с помощью мастера, объединяющую сведения сразу из двух таблиц («Студенты» и «Успеваемость»), многотабличный запрос или отчет.

Установка связей позволяет связать несколько таблиц в одну базу данных.

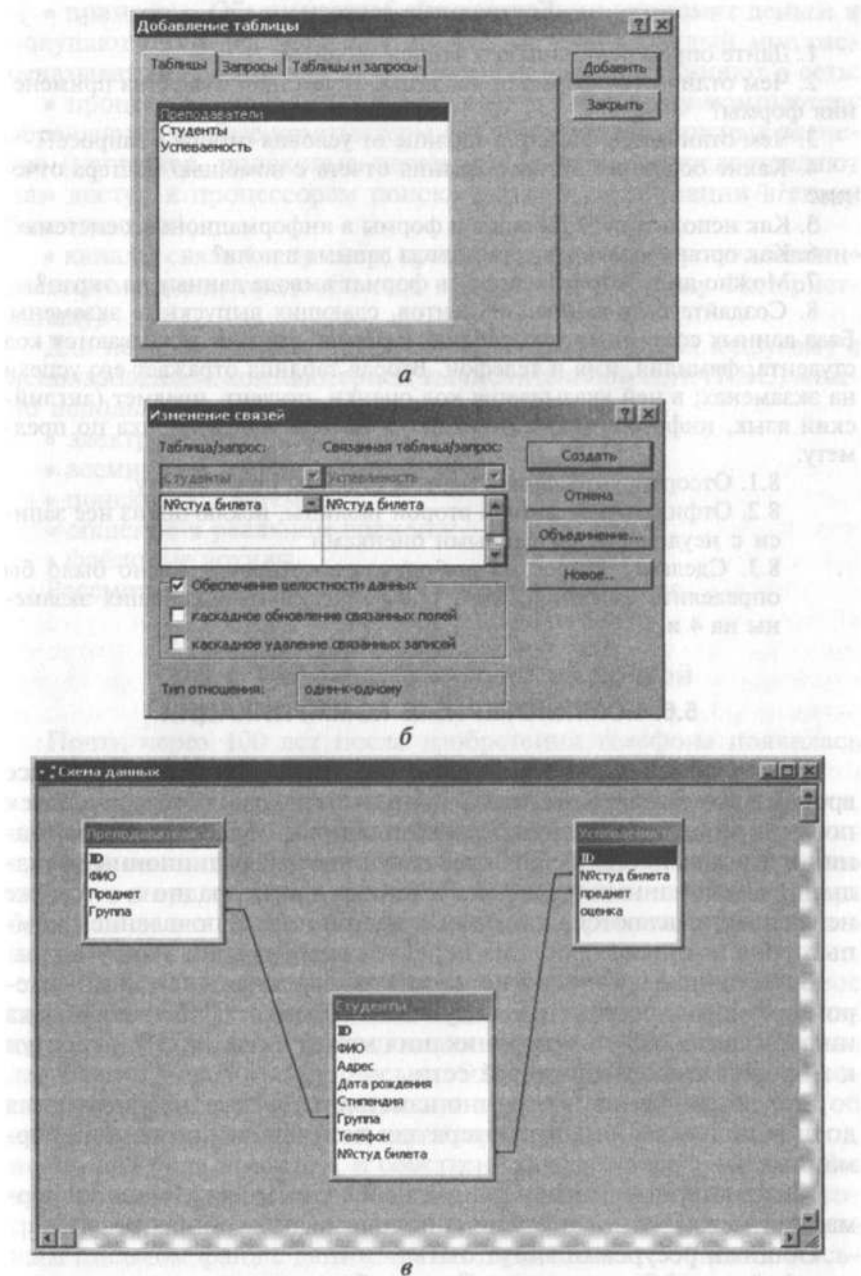


Рис. 5.34. Добавление связей:
 а — добавление таблицы; б — установка характеристик связи; в — окно *Схема данных*

Контрольные вопросы

1. Дайте определение полю и записи таблицы.
2. Чем отличается форма от таблицы? В чем преимущества применения формы?
3. Чем отличается фильтр в таблице от условия отбора в запросе?
4. Какие основные этапы создания отчета с помощью мастера отчетов?
5. Как используются таблицы и формы в информационных системах?
6. Как организовать проверку ввода данных в поля?
7. Можно ли в запросе изменять формат вывода данных на экран?
8. Создайте базу данных студентов, сдающих выпускные экзамены. База данных состоит из двух таблиц. В первой таблице указываются код студента, фамилия, имя и телефон. Вторая таблица отражает его успехи на экзаменах; в ней указывается код оценки, студент, предмет (английский язык, информатика, математика и литература) и оценка по предмету.
 - 8.1. Отсортируйте записи первой таблицы по алфавиту.
 - 8.2. Отфильтруйте записи второй таблицы, исключив из нее записи с неудовлетворительными оценками.
 - 8.3. Сделайте запрос на выборку, по которому можно было бы определить фамилию, имя, телефон студентов, сдавших экзамены на 4 и 5.

5.6. КОМПЬЮТЕРНЫЕ КОММУНИКАЦИИ

Задача обмена информацией стояла перед человеком во все времена его существования. С появлением компьютеров человек получил мощный инструмент накопления и обработки информации и эти задачи стали еще более насущными. Традиционные каналы передачи данных (телефон, телеграф, почта, радио и т.п.) уже не всегда справляются с потоком информации. С появлением компьютеров возникла проблема передачи информации между ними.

В настоящее время при использовании персональным компьютером время распределяется следующим образом: 60 % — обработка информации; 35 % — коммуникация между людьми; 5 % — доступ к мировой информационной сети.

Это соотношение постоянно изменяется в сторону увеличения доли использования компьютера для получения и обмена информацией.

Компьютерные коммуникации необходимы для обмена информацией между компьютерами и использования общих ресурсов.

Общими ресурсами могут быть:

- файловое пространство. Это наиболее частое использование сетей (например, когда вы через Интернет заходите на какой-либо сайт, при этом просматриваете файлы, хранящиеся на удаленном компьютере);

- принтеры. Обычно в любой организации экономят деньги и покупают один-два принтера на всех, а чтобы каждый мог распечатывать свои документы, все компьютеры объединяют в сеть;

- процессорное время, когда к одному мощному компьютеру обращаются слабые компьютеры для проведения сложных расчетов (например, поисковые системы в Интернете предоставляют нам доступ к процессорам поиска нужной информации в своих базах данных);

- каналы связи (например, провайдер, к которому вы дозваниваетесь из дома, предоставляет вам доступ к своему Интернет-каналу).

Для передачи информации от одного компьютера к другому с использованием компьютерной вычислительной сети (КВС) можно использовать:

- электронную почту;
- всемирную информационную сеть Интернет;
- поисковые системы;
- общение в реальном времени;
- файловые архивы.

Рассмотрим каждый из этих способов подробнее.

5.6.1. Работа с электронной почтой

Почти через 100 лет после изобретения телефона появилась компьютерная сеть Интернет.

Электронная почта (e-mail) — первый из сервисов Интернета, наиболее распространенный и эффективный. Днем рождения электронной почты считается 21 октября 1969 г., когда было передано первое в мире электронное письмо. Оно состояло всего из двух букв. После его передачи один из компьютеров, участвовавших в эксперименте, вышел из строя.

E-mail очень похожа на обычную бумажную почту. Обычное письмо состоит из конверта, на котором написан адрес получателя и стоят штампы почтовых отделений пути следования, и содержимого — собственно письма. Электронное письмо также состоит из заголовков, содержащих служебную информацию (об авторе письма, получателе, пути прохождения по сети и т.д.), играющих роль конверта, и собственно содержимого письма. Вы можете вложить в обычное письмо что-нибудь, например фотографию или рисунок. Аналогично вы можете послать электронным письмом файл с данными. Вы подписываете обычное письмо — можно подписать и электронное письмо. Обычное письмо может не дойти до адресата или дойти слишком поздно, как и электронное письмо. Пересылка обычного письма обходится дешево, но электронная почта — самый дешевый вид связи.

Итак, электронная почта имеет те же достоинства (простота, доступность, возможность переслать нетекстовую информацию, возможность подписать и зашифровать письмо) и недостатки (негарантированное время пересылки, возможность доступа третьих лиц во время пересылки, неинтерактивность), что и обычная почта.

E-mail универсален — множество сетей во всем мире, построенных на совершенно разных принципах и протоколах, могут обмениваться электронными письмами с Интернетом, получая тем самым доступ к прочим его ресурсам. Практически все сервисы Интернета, использующиеся обычно как сервисы прямого доступа (on-line), имеют интерфейс к электронной почте, так что даже если у вас нет доступа к Интернету в режиме on-line, вы можете получать большую часть информации, хранящейся в Интернете, посредством электронной почты.

Как и в других видах компьютерной связи, главным в этой системе является взаимодействие двух программ — почтового сервера и почтового клиента. Программное обеспечение, выполняющее функции почтового сервера, установлено, как правило, на удаленном компьютере. Почтовые клиенты — это программы, которые находятся в распоряжении каждого пользователя, отправляющего или принимающего почту. В качестве таких программ можно использовать специальные почтовые программы или Web-технологии.

Работа с почтовыми программами. Современные почтовые программы настроены дружелюбно к пользователю, т.е. имеют интерфейс в форме меню и диалогов, включают в себя удобные средства для организации хранения пришедших писем, ведение «адресных книжек» — списков наиболее часто используемых адресов.

Кроме того, они позволяют прикрепить к письму, имеющему обычный текстовый вид, картинку или любой другой файл. В действительности такой файл будет закодирован в последовательность букв и включен в письмо, но программа получателя автоматически раскодирует его.

Рассмотрим работу с почтовыми программами.

Формат сообщений, позволяющий посылать с письмом различные файлы, в том числе HTML и данные мультимедиа, называется MIME (Многоцелевые расширения почты Интернета). Формат MIME был предложен в 1991 г. Натаном Боренштейном как расширение для базового формата почтовых сообщений SMTP (Простой протокол передачи почты). Если SMTP позволял передавать только текстовую информацию, то MIME дал людям возможность значительно обогатить язык почты в Интернете.

SMTP определяет правила отправки и получения почтовых сообщений по Интернету. Вследствие некоторой функциональной

ограниченности (разработан он был достаточно давно) SMTP не способен накапливать сообщения на стороне получателя, поэтому при получении почты мы имеем дело с еще одним почтовым протоколом.

Протокол POP3 (Почтовый протокол 3) является протоколом для получения сообщений. В соответствии с ним почта принимается сервером и накапливается на нем. Программа «Почтовый клиент» периодически проверяет почту на сервере и загружает сообщения на локальный компьютер.

Таким образом, мы отправляем почту с помощью SMTP или MIME, а принимаем с помощью POP3. Вот почему в процессе создания учетной записи почты вас просят ввести названия как сервера SMTP, так и сервера POP3. Для получения почты не обязательно используется именно POP3. Существует еще один протокол, который позволяет накапливать сообщения на сервере и загружать их на локальный компьютер. Он называется IMAP (Протокол доступа к сообщениям Интернета).

В чем отличие работы по протоколу IMAP от работы по POP3? IMAP позволяет производить над файлами на почтовом сервере различные действия, в отличие от POP3, который просто передает сообщения на ваш компьютер и удаляет их с сервера. Сообщения в этом случае хранятся на сервере, и на их состояние никак не влияют ваши действия, производимые на локальном компьютере. Например, увидев, что на сервер поступили новые сообщения, вы можете просмотреть их заголовки и, решив, что загружать на компьютер эти сообщения не стоит, удалить их с сервера.

Для любой компьютерной платформы существует большое разнообразие почтовых программ. Все почтовые программы имеют, как правило, одинаковый интерфейс. Это говорит о том, что если вы научитесь работать с одной программой такого рода, то освоение других программ не отнимет у вас много времени и сил. Почтовые программы входят в состав широко распространенных коммуникационных пакетов: Outlook Express входит в Microsoft Internet Explorer, а Netscape Messenger — в Netscape Communicator.

Сначала необходимо настроить почтовую программу на работу с почтовым сервером. Рассмотрим в качестве примера настройку почтовой программы Outlook Express для работы с почтовым сервером компании «Провайдер Интернет». После запуска почтовой программы Outlook Express необходимо создать учетную запись *Почта Интернета*, при помощи которой вы сможете отправлять и принимать электронную почту с почтового сервера данного провайдера.

Учетная запись — это набор данных о пользователе почтового сервера, сервера новостей или службы каталогов. Этот набор может содержать ваше имя, электронный адрес, пароль для подклю-

чения к серверу, имена серверов входящей и исходящей почты и т.д.

Настройка почтовой программы. Для настройки почтовой программы в окне программы Outlook Express введите команду *Сервис/Учетные записи*. Откроется диалоговая панель *Учетные записи Интернета*. Выберите вкладку *Почта*. Теперь надо добавить новую почтовую службу. Нажмите кнопку *Добавить* и выберите из списка тип учетной записи — почту.

Откроется диалоговая панель *Мастер подключения к Интернету*. В поле «Ваше имя» укажите ваше имя. Именно его будет видеть человек, получивший от вас письмо. После ввода имени нажмите кнопку *Далее*.

В появившемся окне в поле «Адрес электронной почты» укажите тот адрес, который вы задали при регистрации подключения у провайдера. Адрес следует указать целиком и именно в том виде, в котором вы его создали. Нажмите кнопку *Далее*.

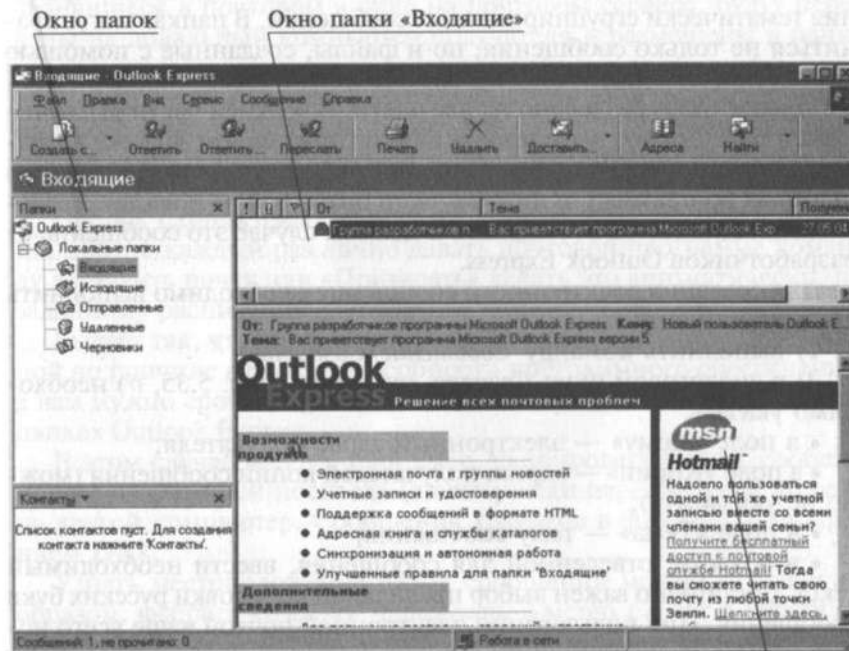
В появившемся диалоговом окне в полях «Сервер для входящей почты» (POP3) и «Сервер для исходящей почты» (SMTP) необходимо указать имена серверов входящей и исходящей почты, которые сообщает провайдер при регистрации подключения (у некоторых провайдеров эти имена совпадают). В поле *Тип сервера* для входящей почты выберите «POP3». Этот протокол наиболее часто используется для электронной почты. Нажмите кнопку *Далее*.

Теперь необходимо указать имя почтового ящика и пароль для входа на почтовый сервер. В поле «Учетная запись POP» впишите имя, которое вы указали при создании своего почтового адреса перед значком «@». Например, если адрес student@mtu-net.ru, то учетная запись будет «student». В поле «Пароль» необходимо указать тот пароль, который был получен при регистрации подключения у провайдера. Нажмите кнопку *Далее*.

Теперь можно отправлять и получать почту, участвовать в группах новостей и пользоваться другими службами Интернета с использованием программы Outlook Express.

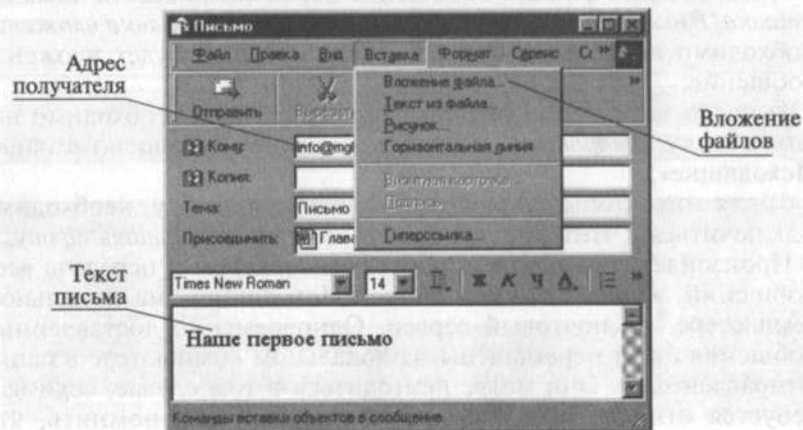
После запуска программы Outlook Express появится окно программы, которое состоит из трех частей (рис. 5.35, а). В левой части окна находится перечень папок, в которых хранится корреспонденция:

- «Входящие» — содержит получаемые адресатом письма;
- «Исходящие» — содержит отправляемые адресатом письма с момента их создания до момента доставки с локального компьютера пользователя на почтовый сервер провайдера;
- «Отправленные» — содержит все письма, доставленные на почтовый сервер;
- «Удаленные» — содержит удаленные письма;
- «Черновики» — содержит заготовки писем.



Окно выделенного сообщения

а



б

Рис. 5.35. Работа с электронной почтой в Outlook Express:
а — окно почтовой программы Outlook Express; б — создание письма в Outlook Express

Пользователь может создавать собственные папки для хранения тематически сгруппированных сообщений. В папках могут храниться не только сообщения, но и файлы, созданные с помощью других приложений.

Правое окно разделено на две части. В верхней части высвечивается содержимое выделенной папки (при первом запуске в папке «Входящие» находятся только два сообщения от разработчиков почтовой программы). В нижней части правого окна показано содержание выделенного сообщения. В данном случае это сообщение от разработчиков Outlook Express.

Для создания электронного сообщения необходимо выполнить следующие шаги:

- 1) выполнить команду *Сообщение/Создать*;
- 2) в диалоговом окне *Создать сообщение* (рис. 5.35, б) необходимо указать:
 - в поле «Кому» — электронный адрес получателя;
 - в поле «Копии» — адреса получателей копии сообщения (можно не указывать);
 - в поле «Тема» — тему сообщения;
 - в области, отведенной для сообщения, ввести необходимый текст. Достаточно важен выбор правильной кодировки русских букв сообщения. При использовании электронной почтой чаще всего используются кодировки Windows или КОИ8-Р. Выбор кодировки осуществляется с помощью команды *Формат/Вид кодировки/Кириллица* (КОИ8-Р/Windows). В сообщение можно вставлять файлы (текстовые, графические, звуковые и т.д.).
- 3) для вставки файла в сообщение необходимо ввести команду *Вставка/Вложение файла*. В появившемся окне *Вставка вложений* необходимо выбрать требуемый файл, который будет вложен в сообщение;
- 4) после завершения работы над сообщением необходимо нажать кнопку *Отправить*, и сообщение будет помещено в папку «Исходящие»;
- 5) для того чтобы отправить сообщение адресату, необходимо подключиться к Интернету и нажать кнопку *Доставить почту*.

Произойдет соединение с почтовым сервером и передача всех сообщений, находящихся в папке «Исходящие» на локальном компьютере, на почтовый сервер. Одновременно доставленные сообщения будут перемещены на локальном компьютере в папку «Отправленные». Они могут пригодиться в том случае, если вам требуется отправить сообщение повторно или вспомнить, что именно и кому вы писали.

Почтовый сервер провайдера передаст сообщения в Интернет, и через некоторое время они окажутся на почтовых серверах получателей. Получатель соединится с Интернетом и произведет операцию доставки почты с почтового сервера провайдера на свой

локальный компьютер. В процессе доставки почты сообщения, хранящиеся в почтовом ящике на почтовом сервере, будут переданы на локальный компьютер получателя и размещены в папке «Входящие».

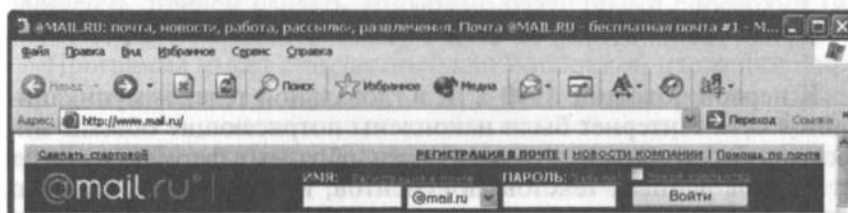
В случае использования кодировки, отличной от примененной при создании сообщения, сообщение будет представлять собой абракадабру. В этом случае необходимо подобрать исходную кодировку с помощью команды *Формат/Вид кодировки*.

Outlook Express предлагает очень гибкую схему доставки почты. Вы можете каждый раз лично давать почтовой программе команду доставить почту или «Приказать» делать это автоматически, по заданному расписанию.

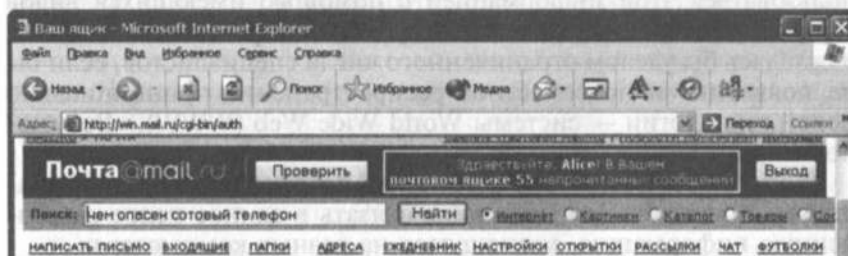
Бывает так, что личная корреспонденция становится недоступной по причине неработоспособности программного обеспечения, а вам нужно срочно прочитать сообщения, хранящиеся в личных папках Outlook Express.

В этом случае можно, например, скопировать файл с сообщениями на сменный носитель (дискету) или переслать его по сети на другой компьютер. Сообщения хранятся в файлах с расширением DBX.

Узнать местоположение папок сообщений можно следующим образом: выбрать команду *Параметры* в меню *Сервис* и щелкнуть по кнопке *Банк сообщений* на вкладке *Обслуживание*. Появится окно



а



б

Рис. 5.36. Web-почта:

а — вход в Web-почту; б — интерфейс Web-почты

Местоположение банка сообщений, в текстовом поле которого указан путь к вашим папкам сообщений.

Электронная почта с Web-интерфейсами. В последнее время для работы с электронной почтой стала использоваться Web-технология. Появились Web-сайты, которые предлагают всем желающим зарегистрировать бесплатный почтовый ящик (например, по адресу: <http://www.mail.ru>).

Преимуществом такой почты является то, что для работы с ней не требуются специальные почтовые программы. Работать с почтой можно с помощью любого браузера после загрузки соответствующей Web-страницы. Все действия выполняются пользователем прямо в окне браузера.

На рис. 5.36, а представлен интерфейс начальной страницы Web-почты. Зарегистрированные пользователи должны ввести свой логин и пароль, после чего они могут войти в почтовую систему.

Для новых пользователей предлагается процедура регистрации. Почтовая система с Web-интерфейсом по своим возможностям аналогична традиционной электронной почте. Сообщения группируются по папкам, можно отправлять сообщения с вложенными файлами, одновременно нескольким абонентам и т.д. (рис. 5.36, б).

Существенной особенностью Web-почты является то, что все сообщения постоянно хранятся на удаленном сервере, а не на локальном компьютере пользователя.

5.6.2. Всемирная информационная сеть Интернет

К первой половине 1990-х гг. в глобальной телекоммуникационной сети Интернет были накоплены потрясающие объемы информации, хранящейся в самых разнообразных формах, начиная от файлов данных, текстов, документов, изображений, звуковых и видеофрагментов и кончая программами. Однако возможность пользоваться этой информацией с помощью имеющихся видов сервиса (электронная почта или протокол передачи данных) так и осталась бы уделом ограниченного числа специалистов, если бы не появление и повсеместное распространение принципиально новой технологии — системы World Wide Web (WWW), «Всемирной паутины».

World Wide Web — это глобальная распределенная информационная система. Она позволила связать в единое целое разрозненную информацию, хранящуюся на разных компьютерах.

WWW — самый популярный и интересный сервис Интернета сегодня, самое популярное и удобное средство работы с информацией. Самое распространенное имя для компьютера в Интернете сегодня — WWW, больше половины потока данных Интернета

приходится на долю WWW, причем скорость роста WWW выше, чем у самой сети Интернет. Сегодня WWW — самая передовая технология Интернета, она уже становится массовой технологией.

Вся информация в этой службе хранится на WWW-серверах в виде гипертекстовых документов, называемых Web-страницами. Эти документы пишутся на языке HTML (Hyper Text Markup Language) и могут содержать информацию различного вида: текст, рисунки, аудио- и видео-, что делает эту информацию чрезвычайно привлекательной для пользователей. Гиперссылки в HTML-документах могут указывать как на другую часть этого документа, так и на другой документ, расположенный на любом сервере сети Интернет. Это позволяет легко отыскивать требуемую информацию, переходя посредством гиперссылок от документа к документу.

Для путешествий по «всемирной паутине» используются специальные программы — браузеры (от *англ.* browse — пролистать, проглядеть, просмотреть). Основная задача программы-браузера — открыть по указанному адресу Web-страницу. Но современные браузеры располагают значительно более широкими возможностями и позволяют работать не только со службой WWW, но и с электронной почтой, телеконференциями и другими службами Интернета.

Одним из наиболее популярных браузеров является Microsoft Internet Explorer, который называют обозревателем. Запустим *Обозреватель*, причем сделать это можно различными способами (из *Главного меню*, с *Панели задач*, с *Рабочего стола*) (рис. 5.37).

Программа имеет стандартный для Windows-приложений фирмы Microsoft вид: в верхней части экрана расположено главное меню, ниже — панель инструментов, под ней — адресная строка, ниже — информационное окно браузера, под ним — информационная строка браузера, показывающая состояние загрузки Web-страницы, в правом верхнем углу — три кнопки управления состоянием и размерами окна программы. В адресное поле программы-браузера (далее — браузера) можно вводить не полный адрес компьютера (URL), а только его часть, начинающуюся с букв *www*. Например, можно вводить не *http://www.aanet.ru*, а *www.aanet.ru*, остальное браузер допишет сам.

Существует два варианта сохранения необходимого адреса:

- 1) выполнить команду *Избранное/Добавить в*;
- 2) щелкнуть по кнопке *Избранное* на панели инструментов.

Если Web-страница долго не открывается (более 3...4 мин), то можно перезагрузить адрес. Для этого надо щелкнуть по кнопке  (Стоп), т. е. прервать загрузку, затем — по кнопке  (*Обновить*). Иногда это приводит к ускорению загрузки страницы.

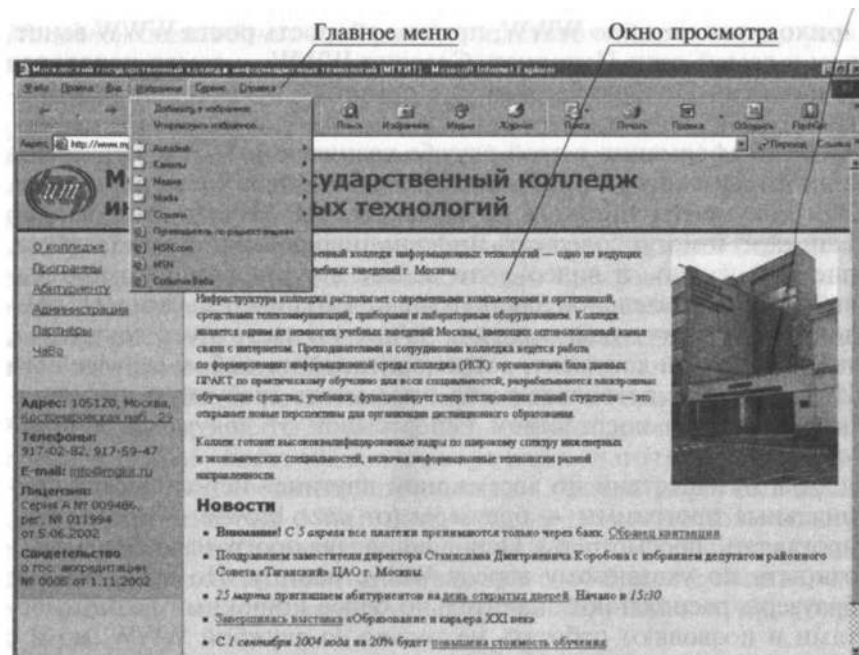


Рис. 5.37. Microsoft Internet Explorer

Указатель мыши в области гиперссылки приобретает вид ладони с указательным пальцем. Открыть документ по адресу, указанному в гиперссылке, можно, нажав один раз левую кнопку мыши на гиперссылке. В результате в текущее окно браузера будет загружен этот документ.

Для открытия документа в новом окне, не закрывая текущее окно, надо нажать правую кнопку мыши (указатель мыши — на гиперссылке) и выполнить команду. Не рекомендуется открывать много окон (больше двух-трех), так как это может привести к замедлению работы программы.

Вернуться на предыдущую страницу можно, щелкнув по кнопке **Назад** на панели инструментов.

Изменить кодировку символов (если на экране появились нечитаемые выражения) можно, выполнив команды **Вид, Шрифты** и выбрав другую (по сравнению с установленной) кодировку. Обычно используется либо кодировка Cyrillic KOI8-R, либо Windows CP-1251.

Рассмотрим технологию работы браузера с WWW-страницами. При загрузке Web-страницы она не только отображается в окне обозревателя, но и помещается в кэш (каталог на диске, в который попадают все когда-либо открываемые страницы), а ее ярлык заносится в специальный журнал посещенных страниц.

При повторном обращении к этой же Web-странице поиск будет осуществляться в кэше и «докачиваться» из Интернета будут только изменения, если в этот момент есть соединение с провайдером (on-line-режим); если соединения нет, то возможен режим автономной работы (off-line) только с информацией из кэша. Если вы при просмотре Web-страниц в режиме on-line открывали по гиперссылкам другие Web-страницы, то и в режиме off-line это будет выполняться. Некоторые Web-страницы могут не открываться. Это значит, что объем Web-страниц, хранящихся в кэше, больше, чем размер дискового пространства, отведенного вами под временные файлы Интернета. Открываться будут только последние страницы, которые вы просматривали и суммарный объем которых не превышает размер временных файлов Интернета.

Для сохранения Web-страниц надо выполнить команду *Файл/Сохранить как*, затем в соответствующем поле ввести имя сохраняемого файла, выбрать папку, в которой хотите сохранить этот файл, и тип файла, в котором хотите сохранить информацию, а затем щелкнуть по кнопке *Сохранить*. Сохранять файл можно в двух форматах: HTML или текстовом. Для выбора типа файла надо нажать на значок черного треугольника в правой части поля «Тип файла» и щелкнуть курсором по нужному формату. Для сохранения рисунков с Web-страниц надо навести указатель мыши на рисунок, щелкнуть правой кнопкой и выполнить команду *Сохранить рисунок как*, а далее выполнить действия, аналогичные указанным в предыдущем абзаце, т.е. задать имя, задать тип файла и указать, в какой папке следует сохранить рисунок.

HTML представляет собой совокупность достаточно простых команд, которые вставляются в исходный текст документа (ASCII-файл) и позволяют управлять представлением этого документа на экране дисплея. Таким образом, текст, подготовленный в любом текстовом редакторе и сохраненный в обычном ASCII-формате, становится Web-страницей (HTML-документом) после добавления в него ряда команд языка HTML. HTML позволяет определять структуру электронного документа с полиграфическим уровнем оформления; результирующий документ может содержать самые разнообразные элементы: иллюстрации, аудио- и видеофрагменты и т.д. Язык включает в свой состав развитые средства для специфицирования нескольких уровней заголовков, шрифтовых выделений, различных групп объектов (например, словари, каталоги или меню для размещения иллюстраций и других фрагментов и др.).

Команды HTML задаются с помощью специальных элементов, называемых тэгами. Тэги (tag) позволяют управлять представлением информации на экране при отображении HTML-документов браузером.

HTML-документ — это просто текстовый файл с расширением *.html (Unix-системы могут содержать файлы с расширением *.html). Вот самый простой HTML-документ:

```
<html>
<head>
<title>
Пример 1
</title>
</head>
<body>
<H1>
Привет!
</H1>
<P>
```

Это простейший пример HTML-документа.

```
</P>
<P>
```

Этот *.html-файл может быть одновременно открыт и в Notepad, и в браузере. Сохранив изменения в Notepad, просто нажмите кнопку *Обновить* в браузере, чтобы увидеть эти изменения реализованными в HTML-документе.

```
<P>
</body>
</html>
```

Для удобства чтения введены дополнительные отступы, однако в HTML это совсем не обязательно. Более того, браузеры просто игнорируют символы конца строки и множественные пробелы в HTML-файлах. Поэтому наш пример вполне мог бы выглядеть и вот так:

```
<html>
<head>
<title>Пример 1</title>
</head>
<body>
<H1>Привет!</H1>
<P>Это простейший пример HTML-документа.</P>
```

<P> Этот *.html-файл может быть одновременно открыт и в Notepad, и в браузере. Сохранив изменения в Notepad, просто нажмите кнопку *Обновить* в браузере, чтобы увидеть эти изменения реализованными в HTML-документе. </P>

```
</body>
</html>
```

Как видно из примера, вся информация о форматировании документа сосредоточена в его фрагментах, заключенных между знаками «<» и «>». Такой фрагмент (например, <html>) называется меткой.

Большинство HTML-меток — парные, т.е. на каждую открывающую метку вида `<tag>` есть закрывающая метка вида `</tag>` с тем же именем, но с добавлением «/».

Метки можно вводить как большими, так и маленькими буквами. Например, метки `<body>`, `<BODY>` и `<Body>` будут восприняты браузером одинаково.

Многие метки помимо имени могут содержать атрибуты — элементы, дающие дополнительную информацию о том, как браузер должен обработать текущую метку. В нашем простейшем документе, однако, нет ни одного атрибута.

Обязательные метки:

- `<html>...</html>`

Метка `<html>` должна открывать HTML-документ. Метка `</html>` должна завершать HTML-документ;

- `<head>...</head>`

Эта пара меток указывает на начало и конец заголовка документа. Помимо наименования документа в этот раздел может включаться множество служебной информации;

- `<title>...</title>`

Все, что находится между метками `<title>` и `</title>`, толкуется браузером как название документа. Netscape Navigator, например, показывает название текущего документа в заголовке окна и печатает его в левом верхнем углу каждой страницы при выводе на принтер. Рекомендуется название не длиннее 64 символов;

- `<body>...</body>`

Эта пара меток указывает на начало и конец тела HTML-документа, которое и определяет содержание документа;

- `<H1>...</H1>` — `<H6>...</H6>`

Метки вида `<Hi>` (где *i* — цифра от 1 до 6) описывают заголовки шести различных уровней. Заголовок первого уровня — самый крупный, шестого уровня — самый мелкий;

- `<P>...</P>`

Такая пара меток описывает абзац. Все, что заключено между `<P>` и `</P>`, воспринимается как один абзац.

Метки `<Hi>` и `<P>` могут содержать дополнительный атрибут `ALIGN` (выравнивать), например:

`<H1 ALIGN=CENTER>`Выравнивание заголовка по центру`</H1>`
или

`<P ALIGN=RIGHT>`Образец абзаца с выравниванием по правому краю`</P>`

Подытожим все, что мы знаем, с помощью следующего примера:

```
<html>
<head>
<title>Пример 2</title>
</head>
```

```

<body>
<H1 ALIGN=CENTER>Привет!</H1>
<H2>Это чуть более сложный пример HTML-документа</H2>
<P>Теперь мы знаем, что абзац можно выравнивать не только
влево, </P>
<P ALIGN=CENTER>но и по центру</P> <P ALIGN=
RIGHT>или по правому краю.</P>
</body>
</html>

```

Теперь вы знаете достаточно, чтобы создавать простые HTML-документы самостоятельно от начала до конца.

5.6.3. Поисковые системы

Если для путешествия по «всемирной паутине» необходим браузер, то для поиска необходимой информации в Интернете используются специальные информационно-поисковые системы, которые размещаются на общедоступных серверах.

Поисковые системы обычно состоят из трех компонентов:

- 1) агент (паук), который перемещается по сети и собирает информацию;
- 2) база данных, которая содержит всю информацию, собираемую пауками;
- 3) поисковый механизм, который люди используют как интерфейс для взаимодействия с базой данных.

Средства поиска и структурирования, иногда называемые поисковыми механизмами, используются для того, чтобы помочь людям найти информацию, в которой они нуждаются. Средства поиска используются для сбора информации о документах, находящихся в сети Интернет. Это специальные программы, которые занимаются поиском страниц в сети, извлекают гипертекстовые ссылки на этих страницах и автоматически сортируют информацию, которую они находят для построения базы данных. Каждый поисковый механизм имеет собственный набор правил, определяющих, как собирать документы. Некоторые следуют за каждой ссылкой на каждой найденной странице и затем исследуют каждую ссылку на каждой из новых страниц и т.д. Некоторые игнорируют ссылки, которые ведут к графическим и звуковым файлам, файлам мультимедиа; другие игнорируют ссылки к ресурсам типа баз данных; третьи проинструктированы, что нужно просматривать прежде всего наиболее популярные страницы.

Когда кто-либо хочет найти информацию, доступную в Интернете, он посещает страницу поисковой системы и заполняет форму, детализирующую информацию, которая ему необходима.

Здесь могут использоваться ключевые слова, даты и другие критерии. База данных отыскивает предмет запроса и выводит соответствующие документы. Чтобы определить порядок, в котором список документов будет показан, база данных применяет алгоритм ранжирования. Различные поисковые системы используют различные алгоритмы ранжирования:

- частоту цитируемости (как много ссылок на данную страницу ведет с других страниц, зарегистрированных в базе и обрабатываемости);

- число слов запроса в текстовом содержимом документа;
- местоположение искомых слов в документе;
- удельный вес слов в общем числе слов документа. Эти принципы применяются всеми поисковыми системами;

- время — как долго страница находится в базе поискового сервера. Поначалу кажется, что это довольно бессмысленный принцип. Но, в Интернете существует много сайтов, которые живут максимум месяц! Если же сайт существует довольно долго, то это означает, что владелец весьма опытен в данной теме и пользователю больше подойдет сайт, который пару лет вещает миру о правилах поведения за столом, чем тот, который появился неделю назад с этой же темой.

База данных выводит ранжированный подобным образом список документов с HTML и возвращает его человеку, сделавшему запрос. Различные поисковые механизмы также выбирают различные способы показа полученного списка: некоторые показывают только ссылки; другие выводят ссылки с первыми несколькими предложениями, содержащимися в документе или заголовок документа вместе с ссылкой. Когда вы щелкаете на ссылке к одному из документов, который вас интересует, этот документ запрашивается у того сервера, на котором он находится.

Приведем примеры нескольких популярных поисковых систем.

AltaVista. www.altavista.com — одна из старейших поисковых систем в Интернете. Первый Web-индекс был представлен компанией в 1995 г.

Ядро поисковой системы обязано своим рождением странной особенности исследовательской лаборатории компании «Digital Equipment Corp». Сотрудники этой лаборатории зачем-то хранили всю свою электронную переписку за последние 10 лет. Чтобы такое количество информации не просто занимало дисковое пространство, а приносила хоть какую-то пользу, была создана программа для индексирования документов и поиска нужных слов в ворохе «пожелтевшей от времени» электронной корреспонденции. Система получилась настолько удачной, что впоследствии с успехом перекочевала на просторы «всемирной паутины».

Индекс AltaVista содержит документы более чем на 25 языках. Локализованные версии сайта AltaVista располагаются в доменах

20 стран. В область поиска можно включить документы на всех поддерживаемых языках или только в документах на определенном языке, а на специальной странице можно указать несколько языков для поиска на всех выбранных языках одновременно.

Окно поиска Alta Vista содержит следующие элементы (рис. 5.38, а):

- поле ввода запроса. В это поле вводится запрос для поиска интересующей вас информации. Запрос может представлять собой вопрос, утверждение, фразу или просто несколько слов, определяющих предмет поиска;

- меню выбора языка. С помощью раскрывающегося списка выбора языка можно ограничить область поиска только документами, написанными на указанном языке. В настоящее время поддерживаются 25 языков (среди них есть и русский);

- вкладки, ограничивающие область поиска. Вкладки, расположенные над полем запроса, используются для того, чтобы указать системе, где искать интересующую вас информацию. По умолчанию открыта вкладка *Web*. Это значит, что AltaVista будет искать информацию среди всех ресурсов World Wide Web. В зависимости от предмета поиска можно ограничиться поиском только среди изображений (Image), аудио- (Audio) или видеофайлов (Video) в тематических каталогах (Directory) или в разделе новостей (News).

- ссылки. В окне поиска имеются ссылки на раздел справочной информации (Help) и страницу настройки «семейного фильтра» (Family Filter).

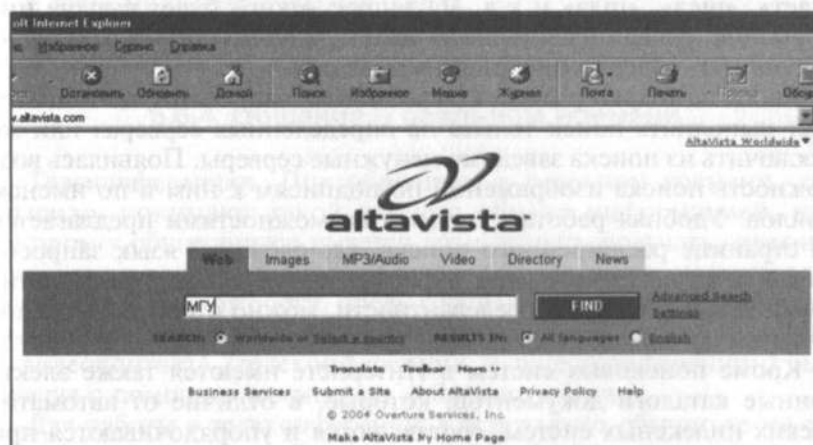
AltaVista является одной из крупнейших систем поиска в Интернете. Она обладает развитым языком запросов и отлично подходит для поиска в сети как специальной информации в определенной области, так и информации более общего характера.

Результат поиска можно видеть на рис. 5.38, б.

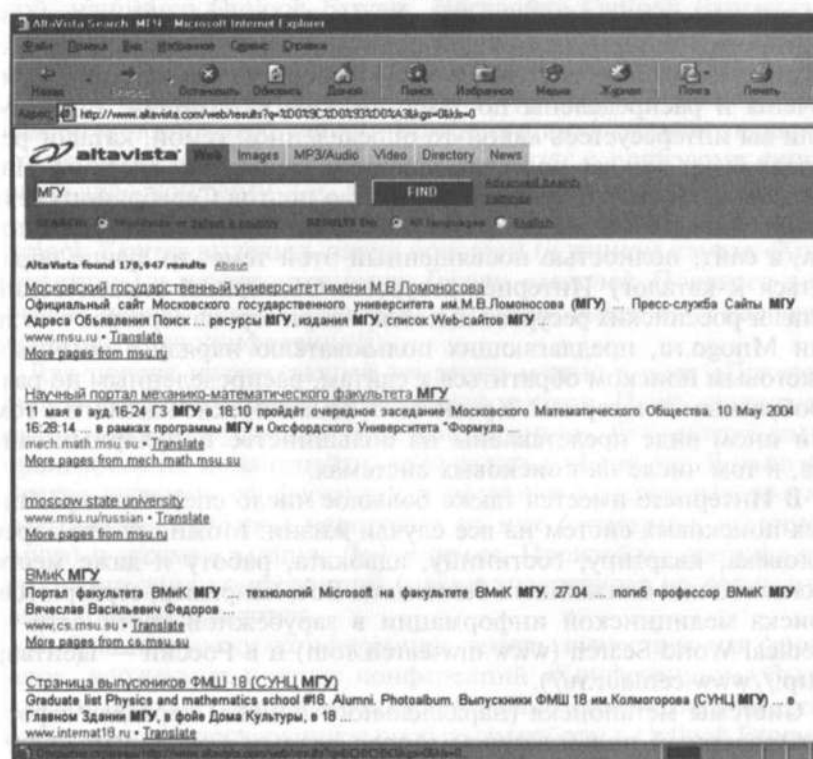
Yandex. Поисковая система Yandex располагается по адресу www.yandex.ru. Она была официально введена в эксплуатацию 23 сентября 1997 г.

Yandex — это полнотекстовая информационно-поисковая система, учитывающая морфологию русского и английского языков. Система Yandex предназначена для поиска информации в электронных текстах различной структуры и разных способов представления (форматов). Yandex не требует от пользователя знания специальных команд для поиска. Достаточно набрать текст: «Где найти дешевые компьютеры?» или «Нужны телефоны Москвы и Московской области» — и вы получите результат: список страниц, где встречаются эти слова.

Независимо от того, в какой форме вы употребили слово в запросе, поиск учитывает все его формы по правилам русского языка. Например, если задан запрос «идти», то в результате поис-



а



б

Рис. 5.38. Поисковая система Alta Vista:

а — начальное окно поиска Alta Vista; б — результат поиска

ка будут найдены ссылки на документы, содержащие слова «идти», «идет», «шел», «шла» и т.д. На запрос «окно» будет выдана информация, содержащая и слово «окон», а на запрос «отзывали» — документы, содержащие слово «отозвали».

Яндекс не только работает с языковыми запросами, но и позволяет выполнять поиск только на определенных серверах или же исключить из поиска заведомо ненужные серверы. Появилась возможность поиска изображений по подписям к ним и по именам файлов. Удобная работа с новыми возможностями предлагается на странице расширенного поиска, где сложный язык запросов сведен к заполнению полей в форме. Кроме стандартной сортировки результатов — по релевантности, можно отсортировать документы по дате обновления.

Кроме поисковых систем в Интернете имеются также электронные каталоги документов, которые, в отличие от автоматических индексных систем, составляются и упорядочиваются при участии человека. Конечно, такие системы содержат информацию о меньшем числе ресурсов, но форма представления и методы сортировки материала выгодно отличают их от большинства индексных поисковых систем. Ресурсы Интернета в каталогах упорядочены и распределены по тематическим категориям, поэтому, если вы интересуетесь какой-то определенной темой, каталог ресурсов будет для вас полезнее текстовой поисковой системы. Например, если вас интересует творчество поэтов Серебряного века и вы хотите найти документы с упоминанием того или иного поэта, а сайт, полностью посвященный этой теме, то лучше обратиться к каталогу Интернет-ресурсов. Обширный тематический каталог российских ресурсов имеется, например, на серверах www.ru или Mnogo.ru, предлагающих пользователю наряду с обычным текстовым поиском обратиться к сайтам, распределенным по разнообразным категориям. Кроме того, тематические каталоги в том или ином виде представлены на большинстве популярных сайтов, в том числе на поисковых системах.

В Интернете имеется также большое число специализированных поисковых систем на все случаи жизни. Можно найти адрес человека, квартиру, гостиницу, адвоката, работу и даже место расположения ближайшего банкомата. Вот, например, серверы для поиска медицинской информации в зарубежной части сети — [Medical World Search \(www.mwsearch.com\)](http://www.mwsearch.com) и в России — [Центавр \(http://www.centaur.ru/\)](http://www.centaur.ru/).

Системы метапоиска (параллельного поиска) передают сформулированный пользователем запрос разным поисковым системам, что позволяет выполнять поиск сразу в нескольких системах. Повторяющиеся ссылки, полученные от разных систем, как правило, исключаются из представляемых пользователю результатов. Вот несколько адресов систем метапоиска:

- 360.ru (www.360.ru);
- Meta (<http://meta-ukraine.com/>);
- MetaCrawler (<http://www.metacrawler.com>).

5.6.4. Общение в реальном времени

Телеконференции. Под общением в реальном времени, как правило, понимают такой процесс обмена информацией, при котором у общающихся имеется возможность получать ответные сообщения с минимальной задержкой. Программы, предоставляющие такие возможности, предназначены для разных целей: просто оповещения о некоторых событиях в сети, установления аудио- и видеоконтакта (телеконференции и видеоконференции) или беседы с помощью клавиатуры (общение в чатах).

Для работы в телеконференциях используют обычно те же самые почтовые программы, что и при работе с электронной почтой, например Outlook Express. Настройка Outlook Express для работы с телеконференциями происходит аналогично настройке на работу с электронной почтой, т.е. создается учетная запись для работы с новостями, например «Конференции». Outlook Express создает одноименную папку «Конференции», которая первоначально пуста. Для того чтобы иметь доступ к почтовому ящику какой-либо конференции, на нее необходимо «подписаться».

Работа с телеконференциями. В окне почтовой программы Outlook Express выделить папку новостей (в данном случае «Конференции») и нажать на кнопку *Группы новостей*. Появится диалоговая панель *Группы новостей*, в которой высветятся названия десятков тысяч конференций.

Для поиска интересующей тематики можно в поле «Показать группы новостей», задать символьный шаблон. Пусть, например, нас интересуют русскоязычные конференции. С большой долей вероятности их можно найти, если задать шаблон *rus*. В окне откроется перечень конференций, в названии которых содержится этот шаблон. Выберем некоторые из них (отметим с помощью мыши) и нажмем кнопку *Подписаться*. Произойдет загрузка содержимого этих конференций (писем участников) на локальный компьютер пользователя.

Выбранные нами конференции теперь существуют в форме папок, вложенных в папку конференций «Конференции». Выделим любую из них, например *microsoft.public.ru.russian.outlookexpress* (конференцию, посвященную проблемам работы в Outlook Express).

В правом верхнем окне мы увидим ее содержание, т.е. перечень присланных в нее сообщений. Выделим интересующее нас сообщение и в правом нижнем окне увидим содержание этого сообщения. Ознакомившись с содержанием телеконференции при же-

лании можно послать в нее собственное сообщение, используя для создания бланка сообщения команду *Сообщение/Ответить в группу новостей*.

Видеоконференции. Другим сервисом, еще только развивающимся, являются видеоконференции. В настоящее время организация видеоконференций требует огромной пропускной способности канала, поэтому широкое распространение они смогут получить только с улучшением качества каналов связи. Видеоконференции представляют собой компьютерный аналог видеотелефона. К компьютеру подключается видеокамера, которая передает изображение пользователя. Это изображение оцифровывается и пересылается всем участникам видеоконференции, которые, находясь даже в разных концах света, смогут во время диалога видеть друг друга.

Чаты. Название этой электронной формы общения говорит само за себя — в переводе с английского языка «chat» означает «дружеский разговор», «беседа». Подавляющее большинство чатов имеет несколько разделов, различающихся по тематике. Наиболее общими являются чаты, носящие, как правило, название «General», или «Общий». Помимо общих чатов существуют еще и тематические. Они могут быть посвящены последним событиям общественно-политической жизни, а также спорту, музыке, человеческим отношениям. Возможно также деление чатов по возрастным группам — при входе посетитель указывает свой возраст или ту возрастную группу, с которой он бы хотел пообщаться.

Чат — это электронная дискуссия, осуществляемая посредством сети Интернет, проходящая в интерактивном режиме, т. е. в режиме реального времени. Интерактивность позволяет участникам дискуссии, находящимся зачастую в различных концах страны, а иногда и земного шара, быстро обмениваться репликами, ведя, таким образом, беседу — диалог.

Каждая реплика участника чата отображается на дисплее в виде отдельной строки, которая начинается с сетевого имени адресанта. На жаргоне пользователей всемирной сети сетевое имя называется «ник» (от *англ. nickname* — прозвище). Затем идет ник адресата, и только после этого — то сообщение, которое адресант направляет адресату. Поскольку в чате может одновременно находиться неограниченное количество посетителей, при отсутствии отсылки к автору сообщения и его получателю (нескольким получателям) может возникнуть путаница.

Для того чтобы облегчить восприятие сообщений, каждому посетителю чата при регистрации предлагается выбрать шрифт и цвет, которыми будут в чате отражаться присылаемые им сообщения. Это значительно облегчает общение в чате, так как зачастую в одном и том же чате посетители разделяются на пары и тройки, внутри которых завязываются собственные дискуссии. При нали-

чий же цветовой дифференциации авторов сообщений и человек, уже долгое время находящийся в чате, и новичок не «потеряется» и легко сможет найти ту часть дискуссии, которая его больше всего интересует. Так как собеседники в чате не видят и не слышат друг друга, цвет сообщений заменяет им лицо и голос: визуальный канал восприятия используется в чате с максимальной эффективностью.

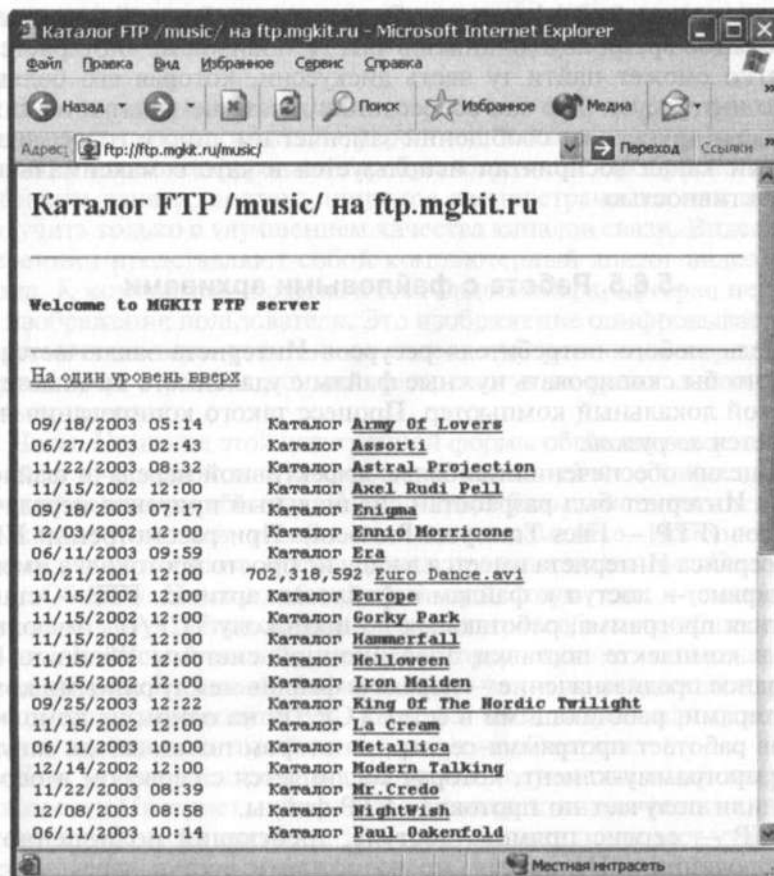
5.6.5. Работа с файловыми архивами

Цель любого потребителя ресурсов Интернета заключается в том, чтобы скопировать нужные файлы с удаленного компьютера на свой локальный компьютер. Процесс такого копирования называется *загрузкой*.

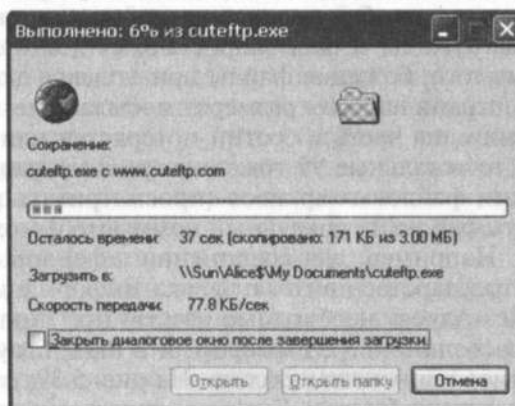
В целях обеспечения наиболее эффективной передачи файлов через Интернет был разработан специальный протокол передачи файлов (FTP — Files Transport Protocol). При рассмотрении FTP как сервиса Интернета имеется в виду не просто протокол, а именно сервис — доступ к файлам в файловых архивах. FTP — стандартная программа, работающая по протоколу TCP/IP, имеющаяся в комплекте поставки операционной системы Windows. Ее исходное предназначение — передача файлов между разными компьютерами, работающими в сетях TCP/IP: на одном из компьютеров работает программа-сервер, на втором пользователь запускает программу-клиент, которая соединяется с сервером и передает или получает по протоколу FTP файлы.

FTP — сервис прямого доступа, требующий полноценного подключения к Интернету, но возможен и доступ через электронную почту — существуют серверы, которые могут прислать вам по электронной почте файлы с любых серверов (anonymous FTP). Однако это может быть весьма неудобно, так как такие серверы сильно загружены и ваш запрос может долго ждать своей очереди. Кроме того, большие файлы при отсылке делятся сервером на части ограниченного размера, посылаемые отдельными порциями. Если одна часть из сотни потеряется или повредится при передаче, то остальные 99 тоже окажутся ненужными. Работать с серверами файловых архивов (просматривать их содержимое и загружать файлы на локальный компьютер) можно с помощью браузера. Например, для соединения с файловым сервером Московского государственного колледжа информационных технологий в поле «Адрес» необходимо ввести: <ftp://ftp.mgkit.ru>.

Произойдет соединение с сервером, и в окне браузера высветится содержимое корневого каталога. На рис. 5.39, а представлено окно обозревателя Internet Explorer, показывающее содержимое каталога по адресу: <ftp://ftp.mgkit.ru/music/>.



a



б

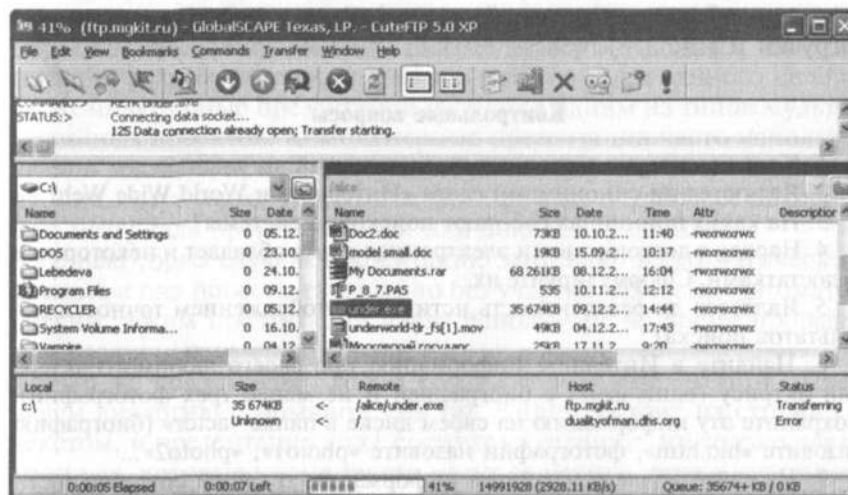


Рис. 5.39. Работа с файловыми архивами:

a — каталог файлового сервера в окне браузера; *б* — загрузка файла; *в* — загрузка файла с помощью FTP-клиента

Для загрузки файла с файлового сервера на локальный компьютер необходимо сначала найти его в иерархической файловой системе. После активизации ссылки на этот файл в открывшемся окне требуется указать папку на локальном компьютере, в которой файл должен быть сохранен. Начнется загрузка файла, процесс которой отображается на информационной панели (скорость передачи, объем загруженной части файла и т.д.) (рис. 5.39. б).

Для работы с серверами файловых архивов часто используются специальные программы, так называемые FTP-клиенты (например, CuteFTP). Такие программы предоставляют пользователю более удобный интерфейс и, самое главное, обеспечивают большую скорость передачи файлов. Важным достоинством специализированных FTP-клиентов является то, что после потери связи с сервером загрузка файла может быть продолжена после восстановления соединения.

В правом окне CuteFTP (рис. 5.39, в) отражается содержимое каталогов FTP-сервера, а в левом окне — каталоги локального компьютера.

Для загрузки выбранного файла (или группы файлов) пользователю достаточно перетащить их с помощью мыши в соответствующий каталог на своем компьютере. В нижней части окна программы CuteFTP отображается подробная информация о процессе загрузки: прошедшее (Elapsed) и оставшееся (Left) время

загрузки, объем загруженной части файла в процентах, скорость загрузки и т.д.

Контрольные вопросы

1. Какие виды компьютерных коммуникаций вы знаете?
2. Являются ли синонимами слова «Интернет» и World Wide Web?
3. На каких принципах работают поисковые системы?
4. Наряду с достоинствами электронная почта обладает и некоторыми недостатками. Сформулируйте их.
5. Является ли релевантность истинным отображением точности результатов поиска?
6. Найдите в Интернете информацию про своего любимого актера или актрису (один файл с биографией и не менее трех фотографий). Сохраните эту информацию на своем диске в папке «actor» (биографию назовите «bio.htm», фотографии назовите «photo1», «photo2»...)
7. На сайте колледжа найдите информацию о вашей специальности (учебный план, преподаватели, выпускники и т.п.). Сохраните эти страницы на своем диске в папке «spec».
8. Заархивируйте папки «actor» и «spec». Архив назовите «inet».
9. Пошлите письмо преподавателю. В качестве темы письма напишите через запятые свою фамилию, имя, группу, и текст «Интернет». В качестве вложенного файла присоедините к письму архив inet.exe.

5.7. МУЛЬТИМЕДИЙНЫЕ ТЕХНОЛОГИИ

Интенсивное развитие аппаратных и программных средств позволило эффективно воспроизводить аудио- и видеoinформацию при помощи компьютера. Звуковое сопровождение работы программы, объяснения и пояснения при помощи программирования речевых роликов, использование анимации и видеороликов постепенно становятся обязательными элементами интерфейса современных прикладных программ.

Мультимедиа (от *англ.* multimedia — многие среды) — это компьютерные интерактивные интегрированные системы, обеспечивающие работу с анимированной компьютерной графикой и текстом, речью и высококачественным звуком, неподвижными изображениями и движущимся видеоизображениями.

Если структурировать информацию, с которой может работать мультимедиа, то можно дать следующее определение.

Мультимедиа — это синтез информации цифрового характера (тексты, графика, анимация), аналоговой информации визуального отображения (видеоизображения, фотографии, картины) и аналоговой информации звука (речь, музыка и другие звуки).

Технологии, использующие мультимедиа, называются мультимедийными. Основная проблема мультимедийных технологий — это

совместная обработка разнородных данных. Она решается за счет перекодировки информации, сжатия/развертки данных, синхронизации разных видов информации при создании единого целого.

Компьютерные презентации являются одним из типов мультимедийных проектов. Компьютерные презентации часто используются в рекламе, при выступлениях на конференциях и совещаниях, они могут также использоваться на уроке в процессе объяснения материала преподавателем или докладов учащихся.

В некоторых случаях презентацию запускают в автоматическом режиме и она повествует о чем-то без участия человека. Автоматический режим презентации часто используют во время проведения различных выставок.

Что же представляет собой компьютерная презентация? Проведем аналогию с обычной книгой. Книга состоит из страниц с текстом, и презентация тоже состоит из страниц, но только электронных, которые кроме текста могут содержать также мультимедийные объекты. Электронные страницы презентации называются *слайдами*.

Книгу мы обычно читаем последовательно, просто перелистывая ее страницы. В процессе просмотра компьютерной презентации могут реализовываться различные последовательности представления слайдов. Для осуществления различных вариантов переходов между слайдами используются либо управляющие кнопки, либо гиперссылки.

Компьютерная презентация представляет собой последовательность слайдов, содержащих мультимедийные объекты: числа, текст, графику, анимацию, видеонизображение и звук. Компьютерные презентации создают с помощью специальной программы — PowerPoint, являющейся приложением MS Office.

PowerPoint позволяет создавать и отображать наборы слайдов, в которых текст сочетается с графическими объектами, картинками, фотографиями, звуком, видеонизображением и даже с мультипликационными спецэффектами. Работу можно перевести на 35-миллиметровые слайды, прозрачные пленки или печатные материалы для выдачи слушателям.

Существуют пять режимов отображения слайдов:

- 1) режим слайдов (выбирается по умолчанию, именно в нем происходит работа с отдельными слайдами);
- 2) режим структуры отображает заголовки и текст слайдов в виде иерархической структуры, напоминающей оглавление книги;
- 3) режим сортировщика слайдов располагает на экране ряд миниатюрных изображений, им удобно пользоваться для просмотра презентации в целом и перестановки слайдов;
- 4) режим страниц заметок — «личный» режим просмотра, в котором каждый из слайдов можно снабдить заметками докладчика;

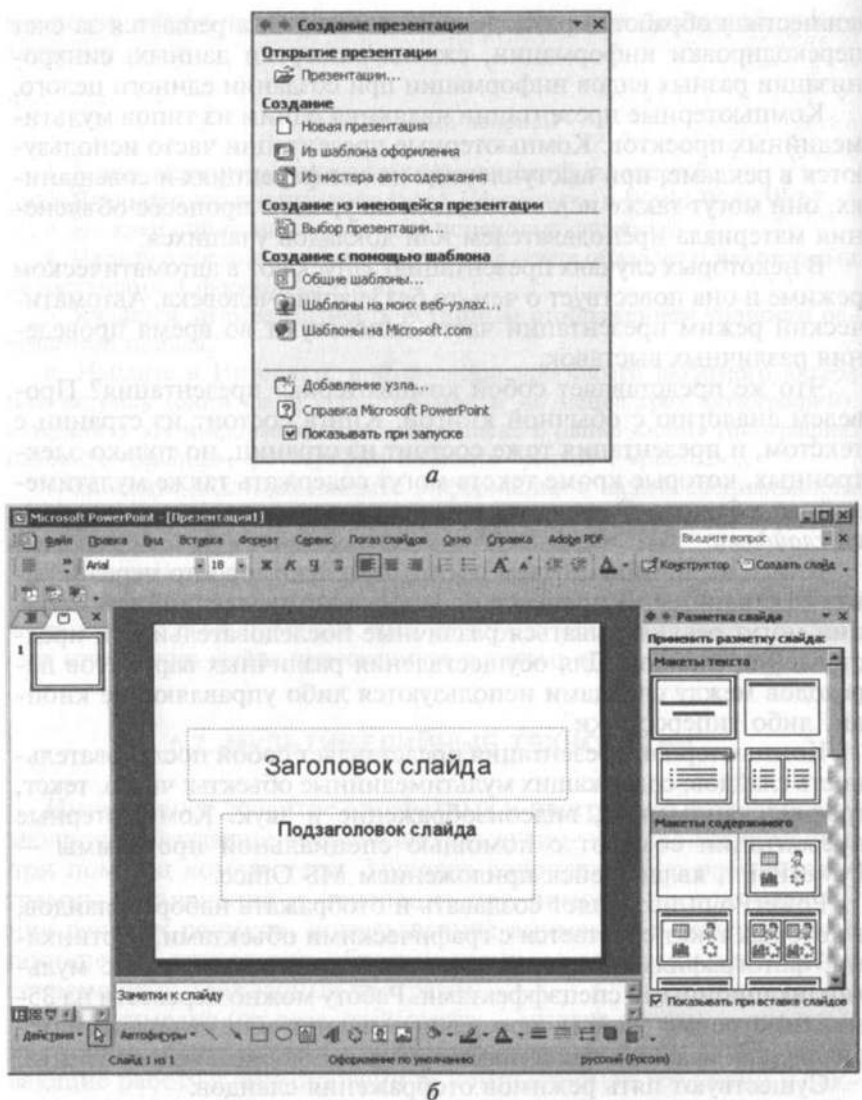


Рис. 5.40. Технология разработки презентации:

a — начало работы; *б* — пустая презентация

5) режим демонстрации используется для просмотра работы и предварительного выполнения готовой презентации.

С чего начать создание презентации? Сначала надо продумать ее проект: сколько будет слайдов в презентации и о чем, в какой последовательности они будут сменять друг друга и т. д.

Шаг 1. При выполнении команды *Создать* в меню *Файл*, открывается несколько возможностей создания презентации (рис. 5.40, а):

Новая презентация открывает пустой холст для создания слайда (рис. 5.40, б); этот вариант предоставляют самую большую свободу действий, но, как и следует ожидать, пользователь должен отчетливо представлять, чего он хочет и как этого добиться.

Шаблон оформления дает возможность применить заранее разработанные структуры, цветовые палитры и фоны для создания набора слайдов по стандартным типам презентации.

Мастер автосодержания — самый легкий способ создания новой презентации; запрашивает у пользователя информацию, после чего создает набор слайдов по указанной теме.

Шаг 2. Ввод и редактирование текста. Создание слайдов.

Слайд может содержать следующие объекты (рис. 5.41):

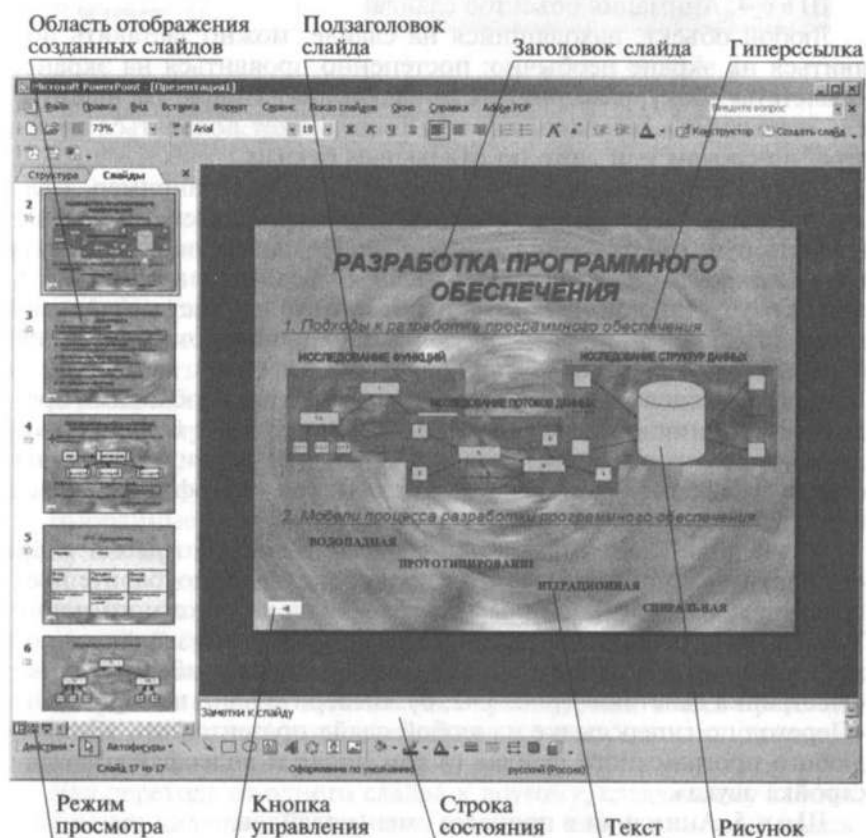


Рис. 5.41. Окно программы PowerPoint

- заголовок, выполненный в виде текста или объекта WordArt;
- форматированный текст (абзацы, колонки, таблицы, списки);
- рисунки;
- гиперссылки.

Технология работы с этими объектами аналогична работе в текстовом процессоре Word. Способ ввода текста в слайд зависит от того, как было создано приложение.

Шаг 3. Выбор дизайна презентации.

Теперь можно выбрать дизайн презентации из коллекции, которая имеется в PowerPoint. Для этого необходимо ввести команду *Формат/Оформление слайда*.

На появившейся диалоговой панели *Шаблоны оформления* можно выбирать различные стили дизайна и просматривать их в окне просмотра. Выбрав подходящий, например «Глобус», нажмите кнопку *Применить*. Все слайды разработанной презентации получат выбранный дизайн.

Шаг 4. Анимация объектов слайда.

Любой объект, находящийся на слайде, можно заставить появиться на экране необычно: постепенно проявиться на экране, «вылететь» сбоку, развернуться до заданного размера, уменьшиться, «вспыхнуть», вращаться и т.д. Текст может появляться целиком, по словам или даже по отдельным буквам.

Для установки параметров анимации объекта (например, *Прямоугольник*) его необходимо выделить, а затем в контекстном меню выбрать пункт «Настройка анимации». Появится диалоговая панель *Настройка анимации* (рис. 5.42, а). Выберите пункт «Добавить эффект» и установите для него необходимые параметры: направление его движения, скорость движения и порядок следования относительно других объектов.

Для переходов между слайдами в презентации необходимо предусмотреть кнопки на слайдах. Щелчок по кнопке будет приводить к переходу на соответствующий слайд. Разместить кнопку на слайде можно через раскрывающийся список «Автофигуры» в левом нижнем углу окна PowerPoint.

Сначала выберем тип кнопки (*Вперед, Назад, Возврат* и т.д.). «Нарисуйте» с помощью мыши кнопку на слайде, подберите цвет и размеры вызовом команды *Автофигуры* из контекстного меню. Теперь можно задать действия, которые будут производиться созданной кнопкой. В контекстном меню кнопки выберите пункт «Настройка действия» (рис. 5.42, б). Выберите один из вариантов: «Переход по гиперссылке на любой слайд презентации», «Запуск любого программного файла» (в том числе и видеофайла), «Настройка звука».

Шаг 5. Анимация в процессе смены слайдов.

Можно создать эффекты анимации при смене одного слайда следующим.

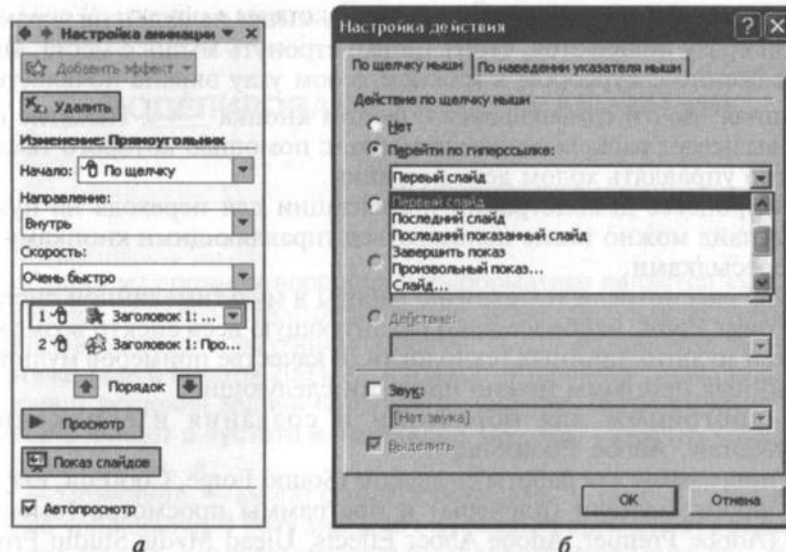


Рис. 5.42. Анимация объекта:

а — настройка анимации; *б* — настройка действия

Для настройки перехода необходимо выделить слайд и дать команду *Показ слайдов/Переход слайда*.

Настройка анимации смены слайдов происходит аналогично шагу 4. Например, в раскрывающемся списке «Эффект» можно выбрать один из типов анимационного эффекта, который будет реализовываться в процессе перехода слайдов.

В раскрывающемся списке «Звуки» можно выбрать звук, которым будет сопровождаться переход слайдов: «Аплодисменты», «Колокольчики», «Пишущая машинка» и т.д. Можно установить любой другой звук, указав на соответствующий звуковой файл.

Выбранные настройки можно применить как к одному текущему слайду, так и сразу ко всем слайдам презентации.

Шаг 6. Демонстрация презентации.

Запуск демонстрации презентации может осуществляться либо командой *Вид/Показ слайдов*, либо нажатием кнопки *Показ слайдов* на панели кнопок. Если делать это с помощью кнопки, то предварительно надо вызвать на экран первый слайд презентации, так как кнопка запускает демонстрацию, начиная с текущего слайда.

Для перехода от одного слайда к другому, следующему за ним, нажимают клавишу [Enter] или щелкают левой кнопкой мыши. Для перемещения по слайдам презентации вперед или назад можно пользоваться клавишей [PageUp] или [PageDown].

В процессе показа слайдов указатель мыши не виден на экране, но он сразу появляется, стоит только стронуть мышь с места. Одновременно с курсором в нижнем левом углу экрана появляется изящная, почти сливающаяся с фоном кнопка . Нажатие на нее вызывает раскрывающееся меню, с помощью которого также можно управлять ходом демонстрации.

В процессе демонстрации презентации для перехода на нужный слайд можно также пользоваться управляющими кнопками и гиперссылками.

Мы рассмотрели технологию работы в мультимедийной системе Power Point, наиболее ярко реализующую весь спектр возможностей мультимедийных технологий. В качестве примеров мультимедийных программ можно привести следующие:

- программы для обработки и создания изображений (CorelDraw, Adobe PhotoShop);
- программы для работы со звуком (Sound Forge, CoolEdit, eJay);
- проигрыватели (плееры) и программы просмотра (вьюеры) (Adobe Premier, Adobe Abber Effects, Ulead Media Studio Pro);
- редакторы трехмерной графики и анимации (3D Studio Max, SoftImage).

Контрольные вопросы

1. Что такое мультимедиа?
2. Перечислите основные функции программы Power Point.
3. Для чего используются гиперссылки в слайдах?
4. В каких режимах можно просматривать презентацию?
5. Сколько объектов изображено на слайде, представленном на рис. 5.41?
6. Разработайте презентацию на следующие темы: «Мой домашний компьютер», «Моя будущая специальность», «Мои друзья», «Мой колледж», «Мир увлечений».

Глава 6

МОДЕЛИРОВАНИЕ И ФОРМАЛИЗАЦИЯ

Одним из сложных вопросов в информатике является «моделирование». Начнем с простого примера. Обыватель формулирует задачу математику следующим образом: «Сколько времени будет падать камень с высоты 200 м?». Математик начинает создавать свой вариант решения задачи приблизительно так: «Будем считать, что камень падает в пустоте и что ускорение силы тяжести составляет $9,8 \text{ м/с}^2$. Тогда...».

— Позвольте, — может сказать обыватель, — меня не устраивает такое упрощение. Я хочу знать точно, сколько времени будет падать камень в реальных условиях, а не в несуществующей пустоте.

— Хорошо, — соглашается математик. — Будем считать, что камень имеет сферическую форму и диаметр s . Какой примерно диаметр?

— Около 5 см. Но он вовсе не сферический, а продолговатый.

— Тогда будем считать, что он имеет форму эллипсоида с полуосями 4, 3 и 3 см и что он падает так, что большая полуось все время остается вертикальной. Давление воздуха примем равным 760 мм рт. ст., отсюда найдем плотность воздуха... s

Если тот, кто поставил задачу, не будет дальше вмешиваться в ход мысли математика, то последний через некоторое время даст численный ответ. Но обыватель может по-прежнему возражать: камень на самом деле вовсе не эллипсоидальный, давление воздуха в том месте и в тот момент не было равно 760 мм рт. ст. и т.д. Что же ответит ему математик?

Он ответит, что точное решение реальной задачи вообще невозможно:

мало того, что форму камня, которая влияет на сопротивление воздуха, невозможно описать никаким математическим уравнением, его вращение в полете также не подвластно математике из-за своей сложности;

воздух не является однородным, так как в результате действия случайных факторов в нем возникают флуктуации колебания плотности;

также нужно учесть, что по закону всемирного тяготения каждое тело действует на каждое другое тело. Отсюда следует, что даже маятник настенных часов изменяет своим движением траекторию падающего камня.

Следовательно, если мы захотим точно исследовать поведение какого-либо предмета, то нам предварительно придется узнать местонахождение и скорость всех остальных предметов Вселенной. А это, разумеется, невозможно.

Чтобы описать явление, необходимо выявить самые существенные его свойства, закономерности, внутренние связи, роль отдельных характеристик явления. Выделив наиболее важные факторы, т. е. создав модель, можно пренебречь менее существенными факторами.

Моделирование имеет чрезвычайно важное значение в человеческой деятельности. Без создания модели невозможно решение задачи на компьютере.

Целью данной главы является рассмотрение вопросов создания моделей, их видов, характеристики информационных моделей, понятия об информационной технологии решения задачи и этапы решения задачи на компьютере.

6.1. МАТЕРИАЛЬНЫЕ И ИНФОРМАЦИОННЫЕ МОДЕЛИ

Модели позволяют представить в наглядной форме объекты и процессы, недоступные для непосредственного восприятия (очень большие или очень маленькие объекты, очень быстрые или очень медленные процессы и др.). Наглядные модели часто используются в процессе обучения. В курсе географии первые представления о планете Земля вы получили, изучая ее модель — глобус; в курсе физики вы изучали работу двигателя внутреннего сгорания по его модели; в курсе химии при изучении строения вещества вы использовали модели молекул и кристаллических решеток; в курсе биологии вы изучили строение человека по анатомическим муляжам и др.

Модели играют чрезвычайно важную роль в проектировании. Без чертежа, схемы, эскиза или макета невозможно изготовить даже простое изделие, не говоря уже о сложных аппаратах и механизмах.

Моделирование — это метод познания, состоящий в создании и исследовании моделей.

6.1.1. Модель

Каждый объект имеет много различных свойств. В процессе построения модели выделяются главные, наиболее существенные для проводимого исследования свойства. В процессе исследования аэродинамических качеств модели самолета в аэродинамической трубе важно, чтобы модель имела геометрическое подобие оригинала, но не важен, например, ее цвет.

Разные науки исследуют объекты и процессы под разными углами зрения и строят различные типы моделей. В физике изучаются процессы взаимодействия и изменения объектов, в химии — их химический состав, в биологии — строение и поведение живых организмов и т.д.

Модель — это новый объект, который отражает существенные особенности изучаемого объекта, явления или процесса.

Любая модель строится и исследуется при определенных допущениях, гипотезах. Модель — результат отображения одной структуры на другую. Отобразив физический объект на математический объект, получим математическую модель физического объекта.

Пример. Рассматривая физическую задачу: тело массой m скатывается по наклонной плоскости с ускорением a под воздействием силы F , Ньютон получил соотношение $F = ma$. Это математическая модель физического объекта. При построении этой модели приняты следующие гипотезы: поверхность идеальна (т.е. коэффициент трения равен нулю); тело находится в вакууме (т.е. сопротивление воздуха равно нулю); масса тела неизменна; тело движется с одинаковым постоянным ускорением в любой точке.

Один и тот же объект может иметь множество моделей, а разные объекты могут описываться одной моделью. Например, чертеж — это графическое изображение различных деталей. Одной моделью описываются динамики эпидемии инфекционного заболевания и радиоактивного распада.

Никакая модель не может заменить сам объект. Но при решении конкретной задачи, когда нас интересуют определенные свойства изучаемого объекта, модель оказывается полезным, а подчас и единственным инструментом исследования.

Один и тот же объект может иметь множество моделей, а разные объекты могут описываться одной моделью.

По назначению модели подразделяются на познавательные, прагматические и инструментальные.

Познавательная модель — форма организации и представления знаний, средство соединения новых и старых знаний. Познавательная модель, как правило, подгоняется под реальность и является теоретической моделью.

Прагматическая модель — средство организации практических действий, рабочего представления целей системы для ее управления.

Реальность подгоняется под некоторую прагматическую модель. Это, как правило, прикладная модель.

Инструментальная модель — средство построения, исследования и/или использования прагматических и/или познавательных моделей.

Познавательные модели отражают существующие, а прагматические — хоть и не существующие, но желаемые и, возможно, исполнимые отношения и связи.

Основные свойства модели:

- конечность — модель отображает оригинал лишь в конечном числе его отношений так как ресурсы моделирования конечны;
- упрощенность — модель отображает только существенные стороны объекта и должна быть проста для исследования или воспроизведения;
- приближительность — действительность отображается моделью оценочно, или приближительно;
- адекватность моделируемой системе — модель должна успешно описывать моделируемую систему;
- наглядность, обозримость основных свойств и отношений;
- доступность и технологичность для исследования или воспроизведения;
- информативность — модель должна содержать достаточную информацию о системе (в рамках гипотез, принятых при построении модели) и давать возможность получать новую информацию.

География, военное дело, судоходство невозможны без информационных моделей поверхности Земли в виде карт. Различные типы географических карт (политические, физические и др.) представляют собой информационные модели, отражающие различные особенности земной поверхности, т.е. одному объекту соответствует несколько моделей.

6.1.2. Формализация

Все модели можно разбить на два больших класса: предметные (материальные) и информационные. Предметные модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме (глобус, анатомические муляжи, модели кристаллических решеток, макеты зданий и сооружений и др.). Информационные модели представляют собой объекты и процессы в образной или знаковой форме. Образные информационные модели (рисунки, фотографии и др.) представляют собой зрительные образы объектов, зафиксированные на каком-либо носителе информации (бумаге, фото- и киноплёнке и др.). Знаковые информационные модели строятся с использованием различных языков (знаковых систем). Знаковая информационная модель может быть представлена в форме текста (например, программы на языке программирования), формулы (например, второго закона Ньютона: $F = ma$), таблицы (например, периодической таблицы элементов Д. И. Менделеева) и т.д.

Если знаковая информационная модель строится с помощью формальных языков, то получается формальная информационная модель (математическая, логическая и др.). Одним из наиболее широко используемых формальных языков является язык математики.

Математическая модель — это система математических соотношений (формул, уравнений, неравенств и т. д.), отражающих существенные свойства объекта или явления.

Язык математики является совокупностью формальных языков: алгебры, тригонометрии, геометрии, теории вероятностей, теории множеств и др. Например, алгебра логики позволяет строить формальные логические модели: формализовать (записать в виде логических выражений) простые и сложные высказывания, выраженные на естественном языке (см. подразд. 2.2).

Процесс построения информационных моделей с помощью формальных языков называется *формализацией*.

Приведем наиболее важные типы моделей с краткими определениями и примерами.

Модель называется *статической*, если среди параметров, участвующих в описании модели, нет временного параметра. Статическая модель в каждый момент времени дает лишь «фотографию» объекта, ее срез.

Пример. Закон Ньютона $F = ma$ — это статическая модель движущейся с ускорением a материальной точки массой m . Эта модель не учитывает изменение ускорения от одной точки к другой.

Модель называется *динамической*, если среди параметров модели есть временной параметр, т. е. она отображает объект во времени.

Пример. Модель $S = gt^2/2$ — динамическая модель, описывающая путь при свободном падении тела.

Модель называется *дискретной*, если она описывает поведение системы только в дискретные моменты времени.

Пример. Если рассматривать только $t = 0, 1, 2, \dots, 10$ (с), то модель $S = gt^2/2$, или числовая последовательность $S(0) = 0$, $S(1) = g/2$, $S(2) = 2g$, $S(3) = 9g/2$, ..., $S(10) = 50g$, может служить дискретной моделью движения свободно падающего тела.

Модель называется *непрерывной*, если она описывает поведение системы для всех моментов времени из некоторого промежутка.

Пример. Модель $S = gt^2/2$, $0 < t < 100$, непрерывна на промежутке времени $(0; 100)$.

Модель называется *имитационной*, если она предназначена для испытания или изучения, проигрывания возможных путей развития и поведения объекта с помощью варьирования некоторых или всех параметров модели.

Пример. Модель экономической системы производства товаров двух видов 1 и 2 в количестве x_1 и x_2 единиц соответственно со стоимостью единиц товара a_1 и a_2 описана соотношением

$$a_1x_1 + a_2x_2 = S,$$

где S — общая стоимость произведенной предприятием всей продукции (видов 1 и 2).

Можно эту модель использовать в качестве имитационной модели, по которой определять (варьировать) общую стоимость S в зависимости от тех или иных значений объемов производимых товаров.

Другим примером имитационной модели может служить старинная русская задача: «Купцу на 100 р. нужно купить 100 голов скота. Сколько можно купить быков, коров и телят, если плата за быка составляет 10 р., за корову — 5 р., за теленка — полтинник (0,5 р).

Модель называется *вероятностной (стохастической)*, если среди параметров, описывающих объект, один или несколько не могут быть определены точно (зависят от случайных событий).

Пример. Если в модели $S = gt^2/2$, $0 < t < 100$, мы учли бы случайный параметр — порыв ветра с силой p при падении тела, то мы получили бы стохастическую модель (уже не свободного) падения.

Модель называется *множественной*, если она представима с помощью некоторых множеств и отношений принадлежности им и между ними.

Пример. Пусть заданы множество $X = \{\text{Николай, Петр, Елена, Екатерина, Михаил, Татьяна}\}$ и отношения: Николай — супруг Елены, Екатерина — супруга Петра, Татьяна — дочь Николая и Елены, Михаил — сын Петра и Екатерины, семьи Николая и Петра дружат друг с другом. Тогда множество X и множество перечисленных отношений Y могут служить множественной моделью двух дружественных семей.

Модель называется *логической*, если она представима логическими функциями.

Пример. Совокупность двух логических функций вида $S = (A \vee \vee B) \& (A \& B)$; $P = A \& B$ может служить математической моделью одноразрядного сумматора.

Модель называется *игровой*, если она описывает, реализует некоторую игровую ситуацию между участниками игры (лицами, коалициями).

Пример. Пусть игрок 1 — добросовестный налоговый инспектор, а игрок 2 — недобросовестный налогоплательщик. Идет процесс (игра) по уклонению от налогов (с одной стороны) и по выявлению сокрытия налогов (с другой стороны). Игроки выбирают натуральные числа i и j , которые можно отождествить соответственно со штрафом, назначаемым игроку 2 за неуплату налогов при обнаружении факта неуплаты игроком 1, и с временной выгодой игрока 2 от сокрытия налогов (в средне- и долгосрочном плане штраф за сокрытие может оказаться более ощутимым). Далее применяется математический аппарат, позволяющий описать игру с учетом стратегий уклонения и поимки.

Модель называется *алгоритмической*, если она описана некоторым алгоритмом или комплексом алгоритмов, определяющим ее функционирование, развитие (см. гл. 7).

Модель называется *визуальной*, если она позволяет визуализировать отношения и связи моделируемой системы, особенно в динамике.

Пример. На экране компьютера часто пользуются визуальной моделью того или иного объекта, например моделью клавиатуры в программе-тренажере по обучению работе на клавиатуре.

Модель называется *геометрической*, *графической*, если она представлена геометрическими образами и объектами.

Примеры. Макет дома является натурной геометрической моделью строящегося дома. Вписанный в окружность многоугольник дает модель окружности. Именно эта модель используется при изображении окружности на экране компьютера. Прямая линия является моделью числовой оси. Параллелограмм часто изображается плоскостью.

6.1.3. Визуализация формальных моделей

В процессе исследования формальных моделей часто производится их визуализация. Для визуализации алгоритмов используются блок-схемы.

Наиболее эффективно математическую модель можно реализовать на компьютере в виде алгоритмической модели. Для этого нужно:

- выделить предположения, на которых будет основываться математическая модель;
- определить, что считать исходными данными и результатами;
- записать математические соотношения, связывающие результаты с исходными данными.

При построении математических моделей далеко не всегда удается найти формулы, явно выражающие искомые величины через данные. В таких случаях используются математические методы, позволяющие дать ответы той или иной степени точности.

Существует не только математическое моделирование, но и визуально-натурное моделирование, которое обеспечивается за счет отображения явлений средствами машинной графики, т.е. перед исследователем демонстрируется своеобразный компьютерный мультфильм, снимаемый в реальном масштабе времени. Так при визуализации формальных физических моделей с помощью анимации может отображаться динамика процесса, производится построение графиков изменения физических величин и т.д. Исследователь может менять начальные условия и параметры протекания процессов и наблюдать изменения в поведении модели.

6.1.4. Модель как совокупность объектов системы

Система является совокупностью взаимосвязанных объектов, которые называются элементами системы (см. подразд. 1.3). Между элементами системы существуют различные связи и отношения (например, ПК является системой, состоящей из различных устройств, которые связаны между собой аппаратно и функционально).

Состояние системы характеризуется ее структурой, т.е. составом и свойствами элементов, их отношениями и связями между собой. Система сохраняет свою целостность под воздействием различных внешних воздействий и внутренних изменений до тех пор, пока она сохраняет неизменной свою структуру. Если структура системы меняется (например, удаляется один из элементов), то система может перестать функционировать как целое. Так, если удалить одно из устройств компьютера (например, процессор), компьютер выйдет из строя, т.е. прекратит свое существование как система.

Любая система существует в пространстве и во времени. Состояние системы в каждый момент времени характеризуется ее структурой, которая может не изменяться (тогда система описывается

Таблица 6.1

Свойства объекта «Дискета»

Объект	Свойства			
	Геометрические размеры	Объем информации	Форматирование	Чтение/запись
1	5"	1,2 Мбайт	Да	Да
2	3,5"	720 Кбайт	Нет	Да
3	3,5"	1,44 Мбайт	Да	Нет

с помощью *статических* моделей) или изменяться со временем (такая система описывается *динамическими* моделями).

В каждый момент времени объект, входящий в состав системы, находится в определенном состоянии, которое характеризуется набором свойств и их значений. Рассмотрим объект «Дискета» системы ПК. Дискету можно охарактеризовать большим числом различных свойств (цвет, геометрические размеры, вес, объем информации и т.д.). Выделим существенные свойства объекта «Дискета» (табл. 6.1).

Некоторые свойства, такие как геометрические размеры и объем информации, имеют постоянное значение, а форматирование может иметь два состояния: отформатированная или неотформатированная (т.е. «да» и «нет»).

Изменение свойств объекта происходит не само по себе, а при выполнении определенной операции или метода.

6.1.5. Типы информационных моделей

Для отражения систем с различными структурами используются различные типы информационных моделей: табличные, иерархические и сетевые.

Табличные информационные модели. Одним из наиболее часто используемых типов информационных моделей является прямоугольная таблица, которая состоит из столбцов и строк. Такой тип моделей применяется для описания ряда объектов, обладающих одинаковыми наборами свойств (см. подразд. 5.5).

В табличной информационной модели перечень однотипных объектов или свойств размещен в первом столбце (или строке) таблицы, а значения их свойств размещаются в следующих столбцах (или строках) таблицы. Табличные информационные модели проще всего строить и исследовать на компьютере с помощью электронных таблиц и систем управления базами данных. Визуализация табличной модели осуществляется, например, путем построения диаграммы в электронных таблицах Excel. Построим таб-

личную информационную модель «Цены устройств компьютера» (табл. 6.2). В первом столбце таблицы будет содержаться перечень однотипных объектов (устройств, входящих в состав компьютера), а во втором и третьем — интересующее нас свойство (например, цена).

Представление объектов и их свойств в форме таблицы часто используется в научных исследованиях. Так, на развитие химии и физики решающее влияние оказало создание Д. И. Менделеевым в конце XIX в. периодической системы элементов, которая представляет собой табличную информационную модель. В этой модели химические элементы располагаются в ячейках таблицы по возрастанию атомных весов, а в столбцах — по количеству валентных электронов, причем по положению в таблице можно определить некоторые физические и химические свойства элементов.

Иерархические информационные модели. В процессе классификации объектов часто строятся информационные модели, которые имеют иерархическую структуру. В биологии весь животный мир рассматривается как иерархическая система (тип, класс, отряд, семейство, род, вид), в информатике используется иерархическая файловая система и т. д.

В иерархической структуре элементы распределяются по уровням — от первого (верхнего) уровня до нижнего (последнего) уровня. На первом уровне может располагаться только один элемент, который является вершиной иерархической структуры. Основное отношение между уровнями состоит в том, что элемент более высокого уровня может состоять из нескольких элементов нижнего уровня; при этом каждый элемент нижнего уровня может входить в состав только одного элемента верхнего уровня.

Граф является удобным способом наглядного представления структуры информационных моделей. Вершины графа отображают элементы системы. Элементы верхнего уровня находятся в отношении «состоять из» к элементам более низкого уровня. Такая

Таблица 6.2

Цены устройств компьютера

Устройство	Цена, у.е.	Цена, р.	Устройство	Цена, у.е.	Цена, р.
Системная плата	80	2400	Монитор	180	5400
Процессор Celeron	70	2100	Звуковая карта	30	900
Память DIMM	15	450	CD-ROM	40	1200
Жесткий диск	130	3900	Корпус	25	750
Дисковод	14	420	Клавиатура	10	300
Видеокарта	30	900	Мышь	5	150



Рис. 6.1. Генеалогическое дерево Рюриковичей (X—XI вв.)

связь между элементами отображается в форме дуги графа (направленной линии в форме стрелки). Если связь между вершинами графа носит несимметричный характер, то такой граф называется *ориентированным*.

Для описания исторического процесса смены поколений семьи используются информационные модели в форме генеалогического дерева. В качестве примера можно рассмотреть фрагмент

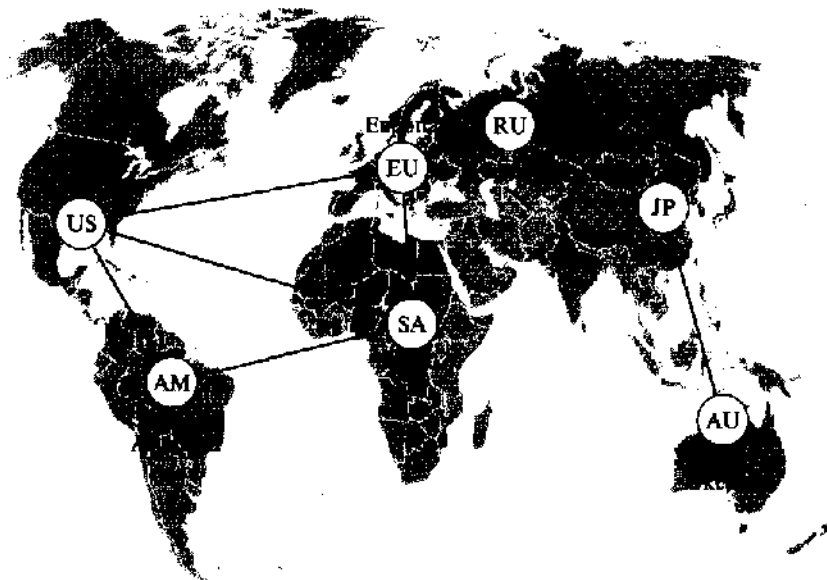


Рис. 6.2. Сетевая структура глобальной сети Интернет

(X—XI вв.) генеалогического дерева династии Рюриковичей (рис. 6.1).

Сетевые информационные модели. Сетевые информационные модели применяются для отражения систем со сложной структурой, в которых связи между элементами имеют произвольный характер. Например, различные региональные части глобальной компьютерной сети Интернет (американская, европейская, российская, австралийская и т.д.) связаны между собой высокоскоростными линиями связи. При этом одни части (например, американская) имеют прямые связи со всеми региональными частями Интернета, а другие могут обмениваться информацией между собой только через американскую часть (например, российская и австралийская).

Построим граф, который отражает структуру глобальной сети Интернет (рис. 6.2). Вершинами графа являются региональные сети. Связи между вершинами носят двусторонний характер и поэтому изображаются ненаправленными линиями, а сам граф называется *неориентированным*.

Контрольные вопросы

1. Приведите примеры, когда один объект характеризуется несколькими моделями.
2. Приведите примеры формальных моделей.
3. Какие системы объектов целесообразно представлять в виде табличных, иерархических, сетевых моделей?
4. Можно ли рассматривать художественное творчество как процесс моделирования?
5. Приведите примеры, когда разные объекты описываются одной и той же моделью.
6. С помощью какой модели — иерархической или сетевой — можно описать классификацию программного обеспечения (см. подразд. 3.3)?

6.2. ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ РЕШЕНИЯ ЗАДАЧ

Роль компьютера для исследования информационных моделей различных объектов и систем необычайно важна.

Основные функции компьютера при моделировании:

- исполнение роли средства подстановки и решения новых задач, не решаемых традиционными средствами, алгоритмами, технологиями;
- исполнение роли средства конструирования компьютерных обучающих и моделирующих сред типа: студент—компьютер—преподаватель, преподаватель—компьютер—студент, преподаватель—компьютер—группа студентов, группа студентов—компьютер—преподаватель, компьютер—студент—компьютер;

- исполнение роли средства моделирования для получения новых знаний;
- «обучение» новых моделей (самообучение моделей).

Компьютерное моделирование — основа представления знаний в вычислительных системах. Компьютерное моделирование для получения новой информации использует любую информацию, которую можно актуализировать с помощью ПК и вычислительных сетей.

Прогресс моделирования связан с разработкой систем компьютерного моделирования, созданием банков моделей, методов и программных систем, позволяющих собирать новые модели из банка моделей.

Вычислительный эксперимент становится новым инструментом, методом научного познания, новой технологией также из-за возрастающей необходимости перехода от исследования линейных математических моделей систем (для которых достаточно хорошо известны или разработаны методы исследования, теория) к исследованию сложных и нелинейных математических моделей систем (анализ которых гораздо сложнее). Вычислительный эксперимент позволяет находить новые закономерности, проверять гипотезы, визуализировать ход событий и т. д.

Процесс разработки моделей и их исследования на компьютере можно разделить на несколько основных этапов.

1. Постановка задачи:

- сбор информации о задаче;
- формулировка условия задачи;
- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных (их типов, диапазонов величин, структуры и т. п.).

На первом этапе исследования объекта или процесса обычно строится описательная информационная модель. Такая модель выделяет существенные с точки зрения целей проводимого исследования параметры объекта, а несущественными параметрами пренебрегает (например, текст задачи в учебнике физики или математики).

2. Анализ и исследование задачи, модели:

- анализ существующих аналогов;
- анализ технических и программных средств;
- разработка математической модели;
- разработка структур данных.

На втором этапе создается формализованная модель, т. е. описательная информационная модель записывается с помощью какого-либо формального языка. В такой модели с помощью формул, уравнений, неравенств фиксируются формальные соотношения между начальными и конечными значениями свойств объектов, а

также накладываются ограничения на допустимые значения этих свойств.

Однако далеко не всегда удается найти формулы, явно выражающие искомые величины через исходные данные. В таких случаях используются приближенные математические методы, позволяющие получать результаты с заданной точностью.

3. Построение компьютерной модели.

На третьем этапе необходимо формализованную информационную модель преобразовать в компьютерную модель, т. е. выразить ее на понятном для компьютера языке, однако не привязывать к конкретному языку программирования.

Существуют два принципиально различных пути построения компьютерной модели:

- построение алгоритма решения задачи и его кодирование на одном из языков программирования;
- построение компьютерной модели с использованием одного из приложений (электронных таблиц, СУБД и др.).

Разработка алгоритма:

- выбор метода проектирования алгоритма;
- выбор формы записи алгоритма (блок-схема, псевдокод и др.);
- выбор тестов и метода тестирования;
- проектирование алгоритма.

Алгоритмическая модель универсальна и не зависит от языка программирования, на котором она в дальнейшем будет реализована.

Примером использования одного из приложений является разработанная нами информационная система «Колледж» (см. подразд. 5.5), которая относится к типу информационно-справочных систем. Ее задача — хранение информации, а также обслуживание простых запросов: на выборку, добавление, удаление и изменение данных и подготовка отчетов. Такая ИС не участвует в принятии решений, она лишь информирует.

Другой тип систем (см. подразд. 1.3) — автоматизированные системы управления. Управление всегда связано с принятием решений. Часто для принятия управляющего решения приходится выполнять довольно сложную обработку данных: производить расчеты, выполнять логический анализ данных.

В управлении и планировании существует целый ряд типовых задач, которые можно переложить на плечи компьютера. Пользователь таких программных средств может даже глубоко не знать математику, стоящую за применяемым аппаратом. Он лишь должен понимать суть решаемой проблемы, готовить и вводить в компьютер исходные данные, интерпретировать полученные результаты. Например, используя приложение MS Office Excel, возможно осуществлять:

- прогнозирование — поиск ответа на вопрос «Что будет через какое-то время?» или «Что будет, если...?»;

- определение влияния одних факторов на другие — поиск ответа на вопрос «Как сильно влияет фактор Б на фактор А?» или «Какой фактор — Б или В — влияет сильнее на фактор А?»;

- поиск оптимальных решений — поиск ответа на вопрос «Как спланировать производство, чтобы достичь оптимального значения некоторого показателя (например, максимума прибыли или минимума расхода электроэнергии)?».

4. Проведение компьютерного эксперимента.

Четвертый этап исследования информационной модели состоит в проведении компьютерного эксперимента. Если компьютерная модель существует в виде программы на одном из языков программирования, то необходимо выполнить:

- выбор языка программирования;
- уточнение способов организации данных;
- запись алгоритма на выбранном языке программирования.

Если компьютерная модель исследуется в приложении, например в электронных таблицах, то можно провести сортировку или поиск данных, построить диаграмму или график и т. д.

5. Анализ результатов.

Пятый этап состоит в анализе полученных результатов и корректировке исследуемой модели. В случае отличия результатов, полученных при исследовании информационной модели, от измеряемых параметров реальных объектов можно сделать вывод, что на предыдущих этапах построения модели были допущены ошибки или неточности. Например, при построении описательной качественной модели могут быть неправильно отобраны существенные свойства объектов, в процессе формализации могут быть допущены ошибки в формулах и т. д. В этих случаях необходимо провести корректировку модели, причем уточнение модели может проводиться многократно, пока анализ результатов не покажет их соответствие изучаемому объекту.

Выполняется анализ результатов решения задачи и уточнение в случае необходимости математической модели с повтором этапов 2—5, перечисленных выше.

Наличие ошибок в только что разработанной программе — это вполне нормальное, закономерное явление. Практически невозможно составить реальную (достаточно сложную) программу без ошибок. Нельзя сделать вывод, что программа правильна, лишь на том основании, что она не отвергнута компьютером и выдала результаты. Ведь все, что достигнуто в данном случае, — получение каких-то результатов, необязательно правильных. В программе при этом может оставаться большое число ошибок.

Отладка программы — это процесс поиска и устранения ошибок в программе, производимый по результатам ее прогона на

компьютере. *Тестирование программы* — это испытание, проверка правильности работы программы в целом либо ее составных частей.

Отладка (от *англ. debugging* — вылавливание «жучков») и тестирование (от *англ. test* — испытание) — это два четко различимых и непохожих друг на друга этапа:

- при отладке происходит локализация и устранение синтаксических ошибок и явных ошибок кодирования;
- в процессе тестирования проверяется работоспособность программы, не содержащей явных ошибок.

Тестирование устанавливает факт наличия ошибок, а отладка выясняет причину. Термин «отладка» появился в 1945 г., когда один из первых компьютеров «Марк-1» прекратил работу из-за того, что в его электрические цепи попал мотылек и заблокировал своими останками одно из тысяч реле машины.

Как бы тщательно ни была отлажена программа, решающим этапом, устанавливающим ее пригодность для работы, является контроль программы по результатам ее выполнения на системе тестов. Программу условно можно считать правильной, если ее запуск для выбранной системы тестовых исходных данных во всех случаях дает правильные результаты.

Но, как справедливо указывал известный теоретик программирования Э. Дейкстра, тестирование может показать лишь наличие ошибок, но не их отсутствие. Нередки случаи, когда новые входные данные вызывают отказ или получение неверных результатов работы программы, которая считалась полностью отлаженной.

Для реализации метода тестов должны быть изготовлены или заранее известны эталонные результаты. Вычислять эталонные результаты нужно обязательно до, а не после получения машинных результатов. В противном случае существует опасность невольной подгонки вычисляемых значений под желаемые, полученные ранее на машине.

Тестовые данные должны обеспечить проверку всех возможных условий возникновения ошибок:

- должна быть испытана каждая ветвь алгоритма;
- очередной тестовый прогон должен контролировать то, что еще не было проверено на предыдущих прогонах;
- первый тест должен быть максимально прост, чтобы проверить, работает ли программа вообще;
- арифметические операции в тестах должны предельно упрощаться для уменьшения объема вычислений;
- число элементов последовательностей, точность для итерационных вычислений, число проходов цикла в тестовых примерах должны задаваться из соображений сокращения объема вычислений;
- минимизация вычислений не должна снижать надежности контроля;

- тестирование должно быть целенаправленным и систематизированным, так как случайный выбор исходных данных приведет к трудностям в определении ручным способом ожидаемых результатов; кроме того, при случайном выборе тестовых данных могут оказаться непроверенными многие ситуации;

- усложнение тестовых данных должно происходить постепенно.

6. Адаптация модели (сопровождение программ).

Сопровождение программ — это работы, связанные с обслуживанием программ в процессе их эксплуатации.

Многочисленное использование разработанной программы для решения задач заданного класса требует проведения дополнительных работ, связанных с доработками программы для решения конкретных задач, проведения дополнительных тестовых расчетов и т.п.

Программа, предназначенная для длительной эксплуатации, должна иметь соответствующую документацию и инструкцию по ее использованию.

Контрольные вопросы

1. Почему невозможно точное исследование поведения объектов или явлений?

2. Какие основные этапы включает в себя решение задач на компьютере?

3. Какие этапы компьютерного решения задач осуществляются без участия компьютера?

4. Что называют математической моделью объекта или явления?

5. Чем тестирование программы отличается от ее отладки?

6. Каким образом программа-отладчик помогает исследовать поведение программы в процессе ее выполнения?

7. На какой стадии работы над программой вычисляются эталонные результаты тестов?

8. В каких случаях могут быть опущены отдельные этапы построения и исследования модели? Приведите примеры создания моделей в процессе обучения.

Глава 7

ОСНОВЫ АЛГОРИТМИЗАЦИИ

На этапе построения компьютерной модели часто встает вопрос разработки алгоритма решения задачи.

Построение алгоритма для задач из какой-либо области может потребовать от человека глубоких знаний в этой области и бывает связано с тщательным анализом поставленной задачи, иногда с громоздкими рассуждениями и сложными расчетами. На поиски алгоритма решения некоторых задач ученые затрачивают многие годы. Но когда алгоритм составлен, решение задачи по готовому алгоритму уже не требует каких-либо рассуждений и сводится только к строгому выполнению команд алгоритма. В этом случае исполнение алгоритма можно поручить не человеку, а компьютеру.

В данной главе рассмотрены основные приемы алгоритмизации, примеры разработки алгоритмов решения типовых задач, даны подробные объяснения методики их решения, приведены схемы алгоритмов.

7.1. АЛГОРИТМ И ЕГО СВОЙСТВА

Возникновение термина «алгоритм» связывают с именем великого узбекского математика IX в. Аль Хорезми, который дал определение правил выполнения основных арифметических операций. В европейских странах его имя трансформировалось в слово «алгорифм», а затем уже в «алгоритм».

В дальнейшем этот термин стали использовать для обозначения совокупности правил, определяющих последовательность действий, выполнение которых приведет к достижению поставленной цели.

Алгоритм — это строгая система правил или инструкций для исполнителя, определяющая некоторую последовательность действий, которая после конечного числа шагов приводит к достижению искомого результата.

Например, инструкция по эвакуации во время пожара; перевод десятичного числа в систему счисления с основанием P .

Каждое действие в записи алгоритма называется *командой*.

В общем случае исполнять алгоритмы может не только человек. Животные, насекомые, растения в процессе жизнедеятельности

выполняют определенные алгоритмы. Поручить исполнение алгоритма можно и неодушевленным механизмам и устройствам.

Если провести анализ, то окажется, что подавляющее большинство действий человек выполняет по определенным алгоритмам, иногда даже не осознавая этого. По определенным рецептам готовятся те или иные кулинарные изделия, по определенным правилам выплавляется сталь, выращивается зерно, решаются математические задачи. Различные справочники в значительной мере являются сборниками алгоритмов, которые представляют собой способы решения тех или иных задач. Можно утверждать, что алгоритм — это способ фиксации и передачи знаний, накопленных человечеством, это богатство культуры, науки и техники.

Исполнитель алгоритма — это абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

Каждый исполнитель может выполнить команды только из некоторого строго заданного списка — системы команд исполнителя (СКИ). Для каждой команды должны быть заданы условия применимости и описаны результаты выполнения команды. Говорят, что исполнитель действует формально, т. е. не отвлекается на содержание поставленной задачи, а только строго выполняя команды. Это очень важная особенность алгоритмов.

Основные свойства алгоритмов:

- **понятность** — исполнителю алгоритма должна быть известна система команд исполнителя;
- **дискретность** (прерывность, раздельность) — алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) команд;
- **определенность** — каждое правило алгоритма должно быть четким и однозначным. Алгоритм должен иметь одно начало и один конец. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче;
- **результативность** (или **конечность**) состоит в том, что алгоритм должен приводить к решению задачи за конечное число шагов;
- **массовость** означает, что алгоритм решения задачи разрабатывается в общем виде, т. е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применения алгоритма.

Способы задания (записи) алгоритмов весьма разнообразны. В частности, можно отметить словесный способ задания алгоритма — на уровне естественного языка; запись музыкальной мелодии в виде нот; графические способы задания алгоритма: чертеж, используемый для изготовления какой-либо детали, маршрут геологической партии, нанесенный на карту, нарисованная

по специальным правилам схема выполнения какой-либо последовательности действий (заметим, что такую схему принято называть блок-схемой алгоритма) и т.д.

Процессор компьютера «понимает» только те алгоритмы, которые записаны в виде двоичных машинных кодов (см. гл. 2). Однако этот естественный для компьютеров, обладающий всеми необходимыми свойствами способ задания алгоритмов очень сложен для использования человеком. Поэтому в информатике применяется ряд специальных способов записи алгоритмов, которые, во-первых, призваны обеспечить соответствие алгоритма всем необходимым требованиям, а во-вторых, приспособлены для их использования как человеком, так и процессором компьютера (после специальной обработки).

Прежде всего вспомним, с какими величинами может работать ПК. Эти величины подразделяются на числовые и текстовые, с одной стороны, и на постоянные (константы) и переменные — с другой.

Назначение переменной — это хранение и обработка данных в алгоритме или программе. То, что переменная может менять свое значение, отличает ее от переменной во многих математических задачах. У переменной есть две характеристики: имя и значение.

Имя переменной — это строка символов, которая отличает эту переменную от других объектов программы. Можно определить имя переменной следующим образом: это такая строка символов, которая идентифицирует (от *англ.* Identify — распознавать, устанавливать идентичность) переменную в алгоритме или программе. Поэтому имена переменных часто называют идентификаторами.

Числовые константы изображаются числами, например: 31,4; 315; 10^8 и т.д. Эти величины не меняют своих значений. *Переменные величины* могут в процессе решения значения задач принимать разные значения. В записи алгоритма они обозначаются буквами, как в математике, или словами: X, Y, MAX, РЕЗУЛЬТАТ. В компьютере для любой величины выделяется ячейка памяти. Если это числовая величина, то в ней хранится число, изображающее значение этой величины.

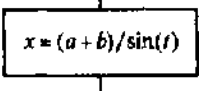
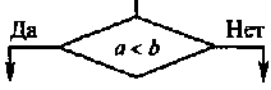
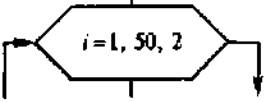
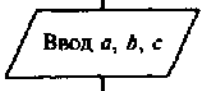
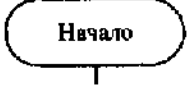
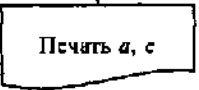
Текстовые величины также могут быть переменными и постоянными. *Текстовые константы* — это любой текст в кавычках: «ХОРОШО», «ВАСЯ», « $Y = 2X(A + B)$ » и т.д. Такие тексты обычно используются для пояснения результатов вычислений, выдаваемых ЭВМ. *Текстовые переменные*, как и числовые, обозначаются буквами или словами. В этом случае в ячейку памяти, выделенную для переменной, помещается не число, а некоторый текст.

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным. При графическом представлении алгоритм изображается в виде пос-

ледовательности связанных между собой функциональных блоков. Такое графическое представление называется схемой алгоритма, или блок-схемой. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий. В табл. 7.1 приведены обозначения, наиболее часто используемые в блок-схемах.

Таблица 7.1

Обозначения, используемые в блок-схемах

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисление по подпрограмме или вспомогательному алгоритму
Ввод-вывод		Ввод-вывод в общем виде
Пуск-остановка		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

Блок «процесс» применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок «решение» используется для обозначения переходов управления по условию. В каждом блоке «решение» должны быть указаны условие (сравнение), которое определяет ход вычислительного процесса.

Блок «модификация» (видоизменение, преобразование) используется для организации циклических конструкций. Внутри блока записывается параметр цикла, для которого указывается его начальное значение, конечное значение и шаг изменения значения параметра цикла для каждого повторения.

Блок «предопределенный процесс» используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращения к библиотечным подпрограммам.

Пример. Разработаем блок-схему алгоритма вычисления площади и периметра прямоугольного треугольника по двум заданным катетам.

Запись на естественном языке:

- 1) ввести значения катетов треугольника: a , b ;
- 2) вычислить площадь по формуле $S = (ab)/2$;
- 3) вычислить значение гипотенузы по формуле $c = \sqrt{a^2 + b^2}$;
- 4) вычислить значение периметра треугольника по формуле $P = a + b + c$;
- 5) вывести полученные значения S и P .

Ту же самую последовательность действий изобразим в виде блок-схемы (рис. 7.1).

Можно сформулировать общие правила построения схемы алгоритма задачи:

- 1) выявить исходные данные, результаты, назначить им имена;
- 2) выбрать метод (порядок) решения задачи;
- 3) разбить метод решения задачи на этапы (с учетом возможностей ПК);
- 4) изобразить каждый этап в виде соответствующей блок-схемы алгоритма и указать стрелками порядок их выполнения;

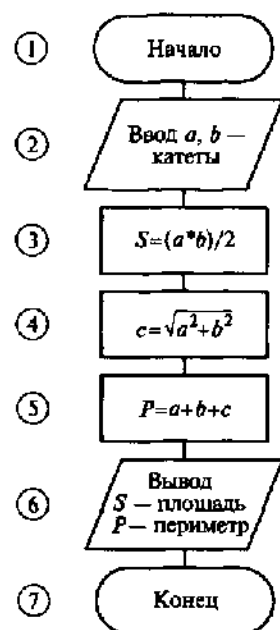


Рис. 7.1. Алгоритм решения задачи

5) в полученной схеме при любом варианте вычислений:

- предусмотреть выдачу результатов или сообщений об их отсутствии;
- обеспечить возможность после выполнения любой операции перехода к блоку «Конец» (к выходу схемы).

Алгоритм можно использовать только в том случае, если он верен, однако на практике в любом достаточно сложном алгоритме (если не принять специальных мер), как правило, содержатся ошибки. Человеку свойственно ошибаться! Это свойственно и составителям алгоритмов, поэтому любой вновь составленный алгоритм нужно отладить.

Отладкой алгоритма называется процесс выявления и исправления ошибок в нем. Суть отладки алгоритма заключается в том, что выбирается некоторый набор исходных данных, называемый тестовым набором (тестом), и задача с этим набором решается дважды: один раз — исполнение алгоритма, второй раз — каким-либо иным способом исходя из условий задачи, «вручную». При совпадении результатов алгоритм считается верным. В качестве тестового набора можно выбрать любые данные, которые позволяют:

- обеспечить проверку выполнения всех операций алгоритма;
- свести число вычислений к минимуму.

Для отладки алгоритма приведенного выше примера зададим значения катетов: $a = 3$, $b = 4$.

Тогда:

в блоке 3 $S = 6$;

в блоке 4 $C = 5$;

в блоке 5 $P = 12$.

Таким образом, простые входные данные показывают, что алгоритм можно использовать для любых катетов прямоугольного треугольника. Тем самым реализуется свойство массовости алгоритма.

Контрольные вопросы

1. Приведите 2—3 примера алгоритмов.
2. Все ли свойства алгоритма должны неукоснительно выполняться и почему?
3. Что такое область применимости алгоритма?
4. Перечислите элементы блок-схем.
5. Приведите примеры использования алгоритмов в физике, математике, литературе, истории.

7.2. БАЗОВЫЕ АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ

Различают алгоритмы линейной, разветвляющейся и циклической структуры, а также алгоритмы со структурой вложенных циклов.

Алгоритмы решения сложных задач могут включать в себя все перечисленные структуры, которые используются для реализации отдельных участков общего алгоритма.

7.2.1. Алгоритмы линейной структуры

Алгоритм линейной структуры — алгоритм, в котором блоки выполняются последовательно друг за другом, в порядке, заданном схемой. Такой порядок выполнения называется естественным (рис. 7.2).

Характерной особенностью каждой структуры является наличие в них одного входа и одного выхода.

Пример. Вычислить высоты треугольника со сторонами a, b, c , пользуясь формулами:

$$h_a = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)};$$

$$h_b = \frac{2}{b} \sqrt{p(p-a)(p-b)(p-c)};$$

$$h_c = \frac{2}{c} \sqrt{p(p-a)(p-b)(p-c)},$$

где $p = (a + b + c)/2$.

Решение задачи состоит из трех этапов:

- 1) ввод исходных данных сторон треугольника a, b, c ;
- 2) расчет по формулам;
- 3) вывод полученных результатов — высот треугольника h_a, h_b, h_c .

При решении данной задачи для исключения повторений используем промежуточную переменную

$$t = 2\sqrt{p(p-a)(p-b)(p-c)}, \text{ тогда } h_a = t/a, h_b = t/b, h_c = t/c.$$

Этот прием значительно сократит количество вычислений.

Алгоритм решения задачи представлен на рис. 7.3.

Обратите внимание, что знак « $\Leftarrow$ » в блоках 3—7 означает операцию присваивания, а не традиционный знак равенства. *Присваивание* означает, что переменной, указанной слева от знака « $\Leftarrow$ », присваивается результат операции или значение переменной, стоящей справа.

В блоке «процесс» слева всегда указывается имя переменной, а справа — ее значение, имя другой переменной или операция.

Примеры операции присваивания:

$$S = 4; S = b; S = (b + 1)/c; S = S + 1.$$



Рис. 7.2. Структура линейного алгоритма

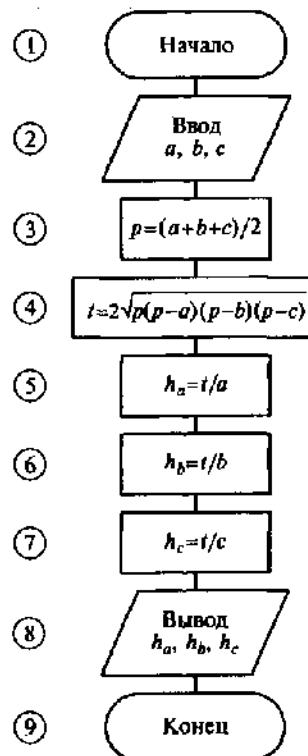


Рис. 7.3. Линейный алгоритм

Последняя операция означает, что значение переменной S должно быть увеличено на 1.

Любая величина сохраняет свое значение, пока ей не будет присвоено новое значение, т.е. чтение числа из ячейки памяти не изменяет содержимого ячейки.

7.2.2. Алгоритмы разветвляющейся структуры

На практике редко удастся представить решение задачи в виде алгоритма линейной структуры. Часто в зависимости от каких-либо промежуточных результатов вычисление осуществляется либо по одним, либо по другим формулам, т.е. в зависимости от выполнения некоторого логического условия вычислительный процесс осуществляется по одной или другой ветви.

Алгоритм такого вычислительного процесса называется *алгоритмом разветвляющейся структуры*. На рис. 7.4 приведены различные варианты алгоритмов разветвляющейся структуры.

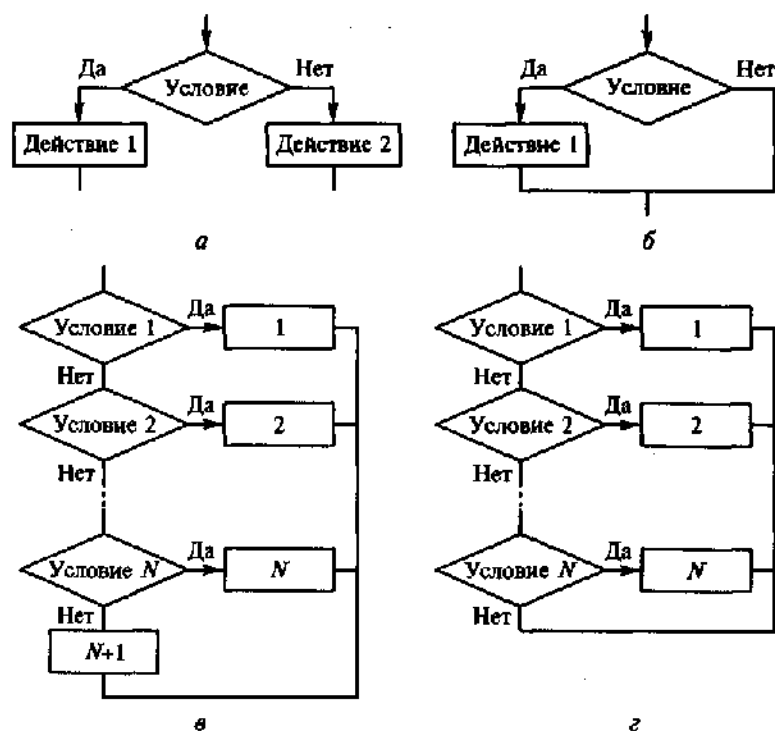


Рис. 7.4. Алгоритмы разветвляющейся структуры:

а — полное ветвление; б — неполное ветвление; в — полный выбор; г — неполный выбор

В качестве условия в разветвляющемся алгоритме может быть использовано любое понятное исполнителю утверждение, которое может соблюдаться (быть истинным) или не соблюдаться (быть ложным). Такое утверждение может быть выражено как словами, так и формулой. Обычно различают два вида условий: простые и составные.

Простым условием называется выражение, составленное из двух арифметических выражений или двух текстовых величин, связанных одним из знаков: < (меньше), <= (меньше или равно), > (больше), >= (больше или равно), = (равно), <> (не равно). Составные условия состоят из двух или более простых, связанных логическими операциями: И, ИЛИ, НЕ.

Например: $x > 10$ И $x < 15$; $a > b$ ИЛИ $c \leq d$.

Условие, содержащееся в логическом блоке, может быть заменено на противоположное:

$$x > 10 \rightarrow x \leq 10;$$

$$k = 15 \rightarrow k \Leftrightarrow 15;$$

$$y < 5 \text{ И } y > 0 \rightarrow y \geq 5 \text{ ИЛИ } y \leq 0.$$

При этом выходы «Да» и «Нет» меняются местами.

Условие можно обозначить логической переменной, значениями которой могут быть только true (истина) или false (ложь),

Таблица 7.2

Примеры логических выражений

Условие	Запись в блок-схеме
Каждое из чисел a , b положительно	$(a > 0) \text{ и } (b > 0)$
Только одно из чисел a , b положительно	$((a > 0) \text{ и } (b \leq 0)) \text{ или } ((a \leq 0) \text{ и } (b > 0))$
Хотя бы одно из чисел a , b , c является отрицательным	$(a < 0) \text{ или } (b < 0), \text{ или } (c < 0)$
Число x удовлетворяет условию $a < x < b$	$(x > a) \text{ и } (x < b)$
Число x имеет значение в диапазоне $[1, 3]$	$(x \geq 1) \text{ и } (x \leq 3)$
Точка (x, y) принадлежит первому или третьему квадранту	$((x > 0) \text{ и } (y > 0)) \text{ или } ((x > 0) \text{ и } (y < 0))$
Точка (x, y) принадлежит внешности единичного круга с центром в начале координат или его второй четверти	$(x^2 + y^2 > 1) \text{ или } ((x^2 + y^2 \leq 1) \text{ и } (x < 0) \text{ и } (y > 0))$
Целые числа a и b являются взаимно противоположными	$a = -b$
Целые числа a и b являются взаимно обратными	$a * b = 1$
Число a больше среднего арифметического положительных чисел b , c , d	$a > (b + c + d)/3$
Хотя бы одна из логических переменных $F1$ и $F2$ имеет значение «Да»	$F1 \text{ или } F2$
Обе логические переменные $F1$ и $F2$ имеют значение «Да»	$F1 \text{ и } F2$
Обе логические переменные $F1$ и $F2$ имеют значение «Нет»	не $F1$ и не $F2$
Логическая переменная $F1$ имеет значение «Да», а логическая переменная $F2$ имеет значение «Нет»	$F1 \text{ и не } F2$
Только одна из логических переменных $F1$ и $F2$ имеет значение «Да»	$(F1 \text{ и не } F2) \text{ или } (F2 \text{ и не } F1)$

например: $F = x > 10$, тогда при $x = 15 F = \text{true}$, а при $x = 5 F = \text{false}$. Логические переменные могут быть связаны логическими операциями И, ИЛИ, НЕ.

Примеры записи логических выражений, истинных при выполнении указанных условий, приведены в табл. 7.2.

Пример 1. Вычислить значения функции

$$z = x^3/y, \text{ где } y = \sin(\pi x) + 0,5.$$

Казалось бы, что решение этой задачи описывает алгоритм линейной структуры. Однако для удовлетворения свойства массовости и результативности алгоритма необходимо, чтобы при любых исходных данных был получен результат или сообщение о том, что задача не может быть решена при заданных данных. Действительно, если $y = 0$, то задача не может быть решена, так как деление на нуль невозможно. Поэтому в алгоритме необходимо предусмотреть условие $y = 0$ и выдать в качестве результата информацию « $y = 0$ ».

Таким образом, вычислительный процесс имеет две ветви. В одной ветви при $y = 0$ необходимо вычислить и вывести значение переменной z , в другой — вывести текст « $y = 0$ ». Такой вычислительный процесс можно описать следующей условной формулой:

$$z = \begin{cases} \text{Вычислить } z = x^3/y, \text{ если } y \neq 0 \\ \text{Вывести «} y = 0 \text{», если } y = 0. \end{cases}$$

Схема алгоритма приведена на рис. 7.5, а.

Пример 2. Упорядочить три числа a , b , c по возрастанию таким образом, чтобы переменной a соответствовало самое маленькое число, b — среднее, c — самое большое.

Для решения данной задачи используем метод перестановок, при котором последовательно сравниваются значения переменных $a \leq b$, затем $a \leq c$ и далее $b \leq c$. Если первое условие $a \leq b$ выполняется, то сразу осуществляется переход к следующему условию $a \leq c$, в противном случае сначала осуществляется перестановка значений соответствующих переменных, а затем переход к следующему условию. Напомним, что в ПК каждая величина хранится в отдельной ячейке памяти, поэтому нельзя сделать перестановку $a = b$; $b = a$, так как в этом случае в обоих переменных a и b окажется значение b . Для перестановки значений двух переменных a и b используется вспомогательная переменная h . Перестановка осуществляется в следующей последовательности: $h = a$, $a = b$, $b = h$.

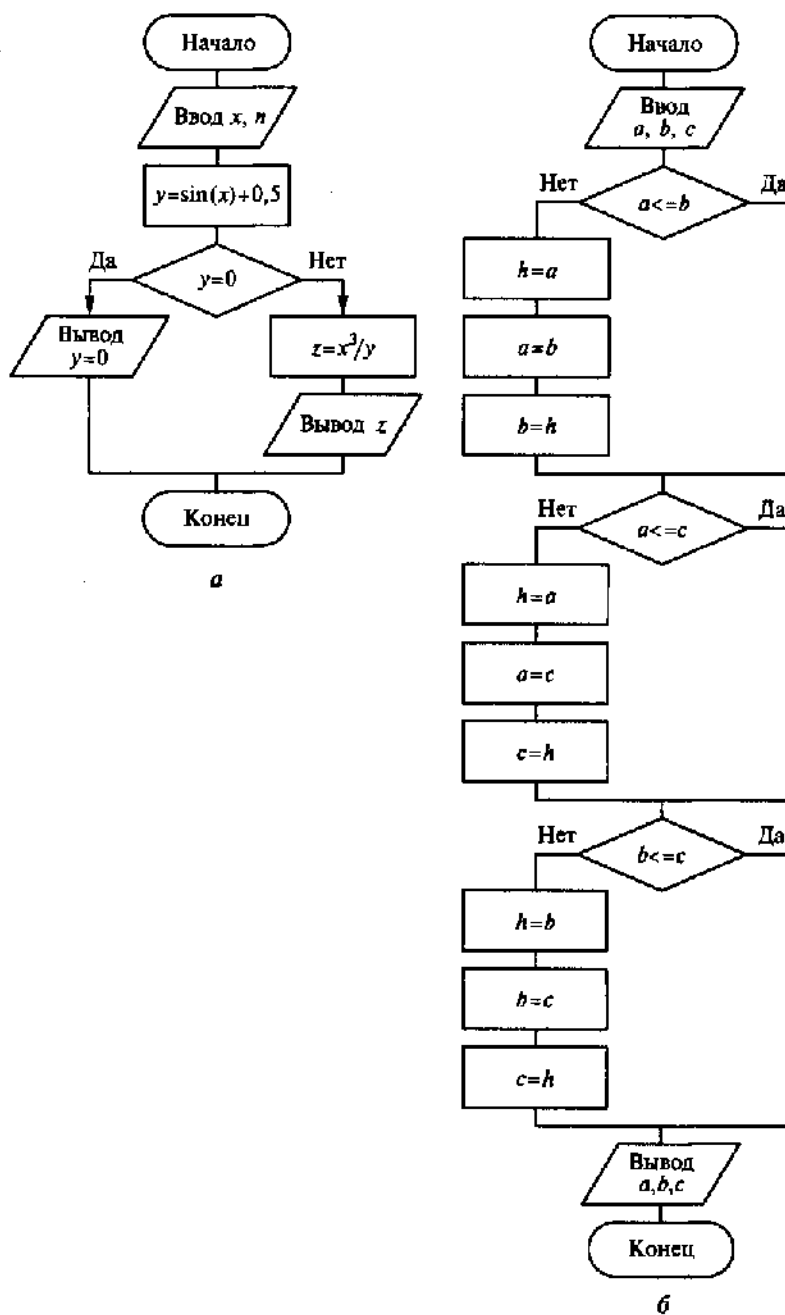


Рис. 7.5. Разветвляющиеся алгоритмы:

а — алгоритм решения задачи, представленной в примере 1; *б* — алгоритм решения задачи, представленной в примере 2

Схема алгоритма представлена на рис. 7.5, б.

Особенность отладки разветвляющихся алгоритмов состоит в следующем: для проверки правильности всех ветвей алгоритма тест должен включать в себя несколько наборов исходных данных (их число должно быть не меньше числа ветвей алгоритма). Проведем отладку для примера 3.

Для полной проверки правильности алгоритма потребуется шесть наборов исходных данных:

<i>a</i>	<i>b</i>	<i>c</i>
1	2	3
1	3	2
2	3	1
2	1	3
3	1	2
3	2	1

Проверьте самостоятельно, что при любом наборе исходных данных результат будет получен одинаковый: $a = 1$, $b = 2$, $c = 3$, т. е. числа будут упорядочены по возрастанию.

7.2.3. Алгоритмы циклической структуры

Часто при решении задач приходится многократно вычислять значения по одним и тем же математическим зависимостям для различных значений входящих в них величин. Такие многократно повторяемые участки вычислительного процесса называются *алгоритмами циклической структуры*, или *циклами*. Использование циклов позволяет существенно сократить объем схемы алгоритма и длину соответствующей ей программы. Различают циклы с заданным и неизвестным числом повторений (рис. 7.6).

Для организации цикла необходимо:

- 1) задать перед циклом начальное значение переменной, изменяющейся в цикле;
- 2) изменять переменную перед каждым новым повторением цикла;
- 3) проверять условие окончания или повторения цикла;
- 4) управлять циклом, т. е. переходить к его началу, если он не закончен, или выходить из него по окончании.

Последние три функции выполняются многократно.

Переменная, изменяющаяся в цикле, называется *параметром цикла*. В одном цикле может быть несколько параметров. Основная сложность в этом процессе — вывести закономерности (формулы) изменения параметров.

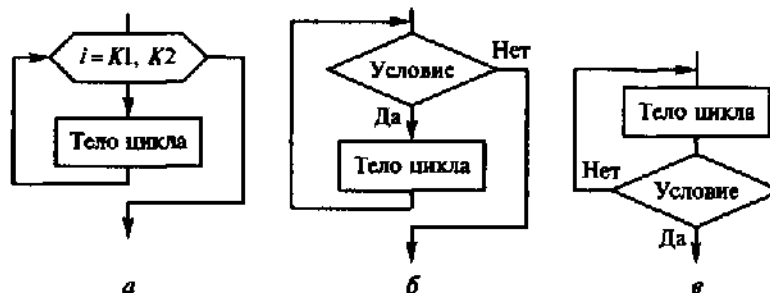


Рис. 7.6. Виды циклов:

a — с заданным числом повторений (арифметический цикл); *б* — с неизвестным числом повторений (цикл с предусловием); *в* — с неизвестным числом повторений (цикл с послусловием)

Пример 1. Вычислить произведение целых нечетных чисел от m до n ($m < n$).

Здесь исходными данными задачи являются величины от m , $m + 2$, $m + 4$, ..., n ; результатом — p .

Метод решения задачи следующий. Последовательно перемножаем m на $m + 2$, полученный результат — на $m + 4$ и т.д. Разобьем указанный процесс на этапы, запишем операцию первого, второго и третьего этапов и исходя из них выведем и представим операцию, выполненную на i -м этапе. Запишем операцию последнего этапа.

$$\text{Этап 1: } p_1 = m(m + 2).$$

$$\text{Этап 2: } p_2 = p_1(m + 4).$$

$$\text{Этап 3: } p_3 = p_2(m + 6).$$

$$\text{Этап } i: p_i = p_{i-1}(m + 2i).$$

$$\text{Этап } r \text{ (последний): } p_r = p_{r-1}n.$$

Формула $p_r = p_{r-1}n$ есть рабочая формула создаваемого циклического алгоритма, в котором задействованы переменные i и p .

Найдем начальное значение переменной p . Представим формулу первого этапа в виде

$$p_1 = p_0(m + 2).$$

При этой замене не должно измениться значение p_1 , а для этого до начала вычислений следует принять $p_0 = m$. Это и есть начальное значение переменной p . Сопоставим операции i -го и r -го этапов, из чего следует условие повторения цикла $m + 2i \leq n$, т.е. $i \leq (n - m)/2$.

Выделим и запишем параметры переменных p и i :

$$p = \begin{cases} m \\ p(m+2i); i = i+1 \\ ? \end{cases} \begin{cases} 1 \\ (n-m)/2. \end{cases}$$

Здесь индексы при p мы опустили, так как для вычисления любого последующего значения переменной p достаточно знать лишь ее предшествующее значение.

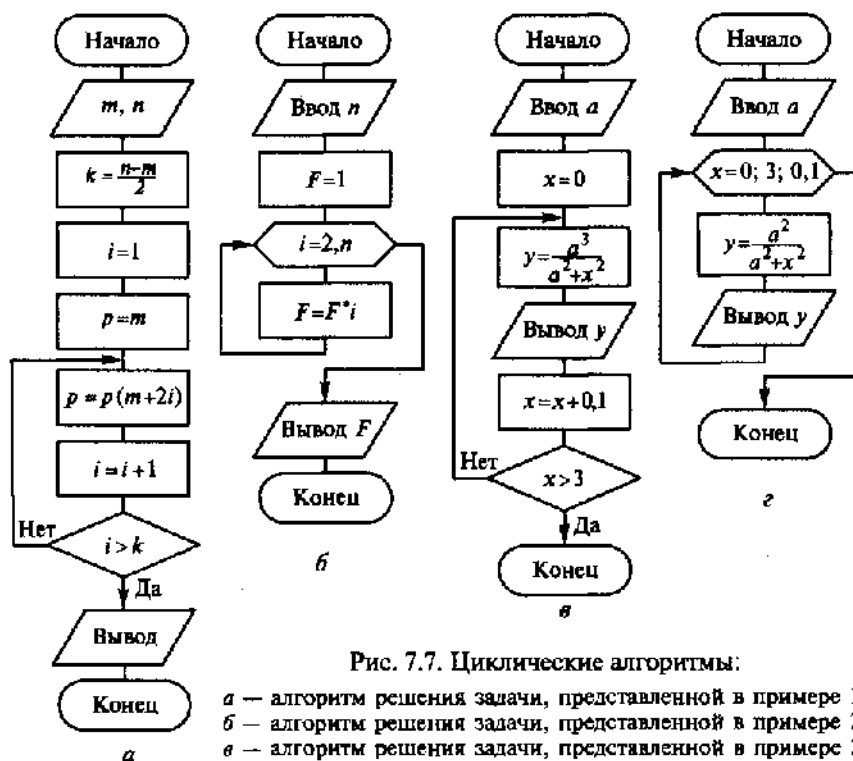
Алгоритм решения задачи представлен на рис. 7.7, а.

Пример 2. Вычислить факториал натурального числа N .

Факториал числа (!) — это произведение всех натуральных чисел от 1 до N :

$$M = 1 * 2 * 3 * \dots * N.$$

По определению $0! = 1$ и $1! = 1$.



Исходными данными является значение N , результат будет получен в переменной F . В переменной F накапливается значение факториала числа. Для реализации вычислений используется арифметический цикл. Параметр цикла i меняется в интервале от 2 до N с шагом 1 (поэтому в данном примере шаг не указан). Выход из цикла произойдет, когда значение параметра цикла превысит значение N . В арифметическом цикле значение параметра цикла увеличивается на значение шага автоматически. Операция $F = F * i$ обеспечивает постепенное накопление произведений в переменной F , причем используется предыдущее значение этой же переменной.

Из примера 1 видно, что перед началом цикла необходимо установить начальное значение переменной F .

Обобщенная формула получения факториала имеет вид

$$F = F * i,$$

т.е. $F_1 = F_0 * 2 = 1! * 2$; $F_2 = F_1 * 3 = 2! * 3$; $F_3 = F_2 * 4 = 3! * 4$ и т.д.

Следовательно, начальное значение F должно быть равно единице.

Блок-схема алгоритма приведена на рисунке 7.7, б.

Пример 3. Вычислить и вывести на печать значения функции

$$y = \frac{a^3}{a^2 + x^2} \text{ при значении } x, \text{ изменяющемся от } 0 \text{ до } 3 \text{ с шагом } 0,1.$$

Перед первым выполнением цикла необходимо задать начальное значение аргумента x , равное 0, а затем вычислить значение y и вывести его на печать. При каждом новом выполнении цикла необходимо изменять аргумент на величину шага, равного 0,1. Чтобы процесс не был бесконечным, необходимо задать условие повторения ($x \leq 3$) или окончания цикла ($x > 3$).

Схема алгоритма (цикл с послеусловием) приведена на рис. 7.7, в. Возможно применение арифметического цикла (рис. 7.7, г).

К циклам с условием относятся *итерационные циклы*, характеризующиеся последовательным приближением к исходному значению с заданной точностью. Особенностью итерационного цикла является то, что число повторений операторов тела цикла заранее неизвестно. Для его организации используется цикл с неизвестным числом повторений (см. рис. 7.6, а, б). Выход из итерационного цикла осуществляется в случае выполнения заданного условия: как правило, достижения установленной точности.

Пример 4. Составим алгоритм вычисления суммы ряда

$$S = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{i-1} \frac{x^i}{i} + \dots$$

с заданной точностью ξ (для данного знакопеременного степенного ряда требуемая точность будет достигнута, когда очередное слагаемое станет по абсолютной величине меньше ξ).

Особенностью нашей задачи является то, что число слагаемых (а следовательно, и число повторений тела цикла) заранее неизвестно. Поэтому выполнение цикла должно завершиться в момент достижения требуемой точности. При составлении алгоритма нужно учесть, что знаки слагаемых чередуются и степень числа x в числителях слагаемых возрастает. Решая эту задачу «в лоб» путем вычисления на каждом i -м шаге частичной суммы и накопления ее в переменной S

$$S = S + (-1)^{i-1} \cdot \frac{x^i}{i},$$

получим очень неэффективный алгоритм, требующий выполнения большого числа операций.

Гораздо лучше организовать вычисления следующим образом: если обозначить числитель какого-либо слагаемого буквой p , то у следующего слагаемого числитель будет равен $-p \cdot x$ (знак « $-$ » обеспечивает чередование знаков слагаемых), а само слагаемое m будет равно p/i , где i — номер слагаемого m — фактически является членом убывающего ряда. Поэтому условие выхода из цикла:

$$|m| \leq \xi$$

(модуль используется в связи с тем, что в исходной формуле существует чередование знаков).

На блок-схеме алгоритма решения задачи используется цикл с предусловием, поэтому на рис. 7.8 приведено условие, обратное рассмотренному ($|m| > \xi$), а выход из цикла осуществляется, когда условие не выполняется.

Алгоритм, в состав которого входит итерационный цикл, называется *итерационным алгоритмом*. Итерационные алгоритмы используются при реализации итерационных численных методов. В итерационных алгоритмах необходимо обеспечить обязательное достижение условия выхода из цикла (сходимость итерационного процесса). В противном случае произойдет заклинивание алгоритма, т.е. не будет выполняться основное свойство алгоритма — результативность.

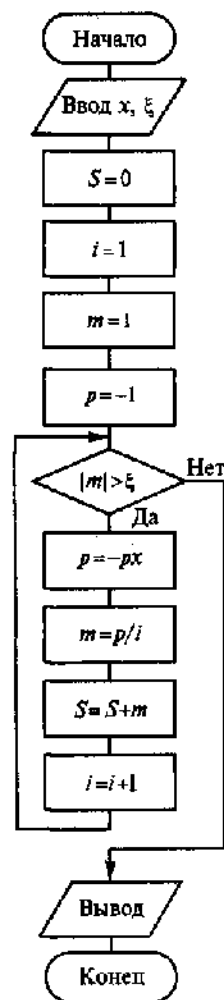


Рис. 7.8. Итерационный алгоритм

7.2.4. Массивы

Часто в технике, науке и повседневной жизни используются не отдельные числа и величины, а множества связанных однородных величин. Так, дата — это совокупность трех значений, например 19.10.88. С листом бумаги связывается два значения — длина и ширина, например 250×170 мм; с чемоданом — три значения — длина, ширина, высота, например $500 \times 300 \times 150$. Многочлен $Y = -7x^4 + 4x^3 + 5x^2 + 2x + 3$ однозначно определяется совокупностью пяти чисел-коэффициентов: 3, 2, 5, 4, -7. Несложно представить себе и другие множества связанных однородных величин с шестью, семью и более элементами. Такие множества широко используются и в информатике, где они называются массивами.

Массивы — последовательности однотипных элементов, число которых фиксировано и которым присвоено одно имя.

Положение элемента в массиве однозначно определяется его индексами (одним — если массив одномерный; несколькими — если массив многомерный). Например, список студентов в группе можно рассматривать как массив (все его элементы — текстовые величины), а номер по журналу — как индекс элемента этого массива.

Список студентов

1. Иванов

2. Киселев

3. Сидоров

...

25. Яковлев

Назовем его массивом А. В этом одномерном массиве 25 элементов:

$A[1] = \text{«Иванов»}$, $A[2] = \text{«Киселев»}$, $A[3] = \text{«Сидоров»}$, ..., $A[25] = \text{«Яковлев»}$.

В алгоритме переменную, являющуюся элементом массива, называют переменной с индексом. Параметром цикла при использовании простой переменной является сама переменная, а при использовании переменной с индексом — ее индекс.

Массивы называют *двумерными*, или матрицами, когда каждый элемент массива снабжается двумя индексами. Двумерные массивы можно представить как таблицу. Индексами элемента будут номер строки и номер столбца, на пересечении которых находится данный элемент. Данные в таблице должны быть одного типа.

Примером двумерного массива может служить ведомость (табл. 7.3).

Оценки студентов можно представить двумерным массивом В, каждый элемент которого будет иметь два индекса (первый будет соответствовать номеру строки, а второй — номеру столбца).

Таблица 7.3

Формирование двумерного массива

ФИО	Математика	Информатика	Иностранный язык
Андреев М. К.	3	4	5
Иванов И. С.	3	3	3
Петров М. П.	4	4	5
Сорокина М. И.	5	4	5
Яшин П. П.	4	3	4

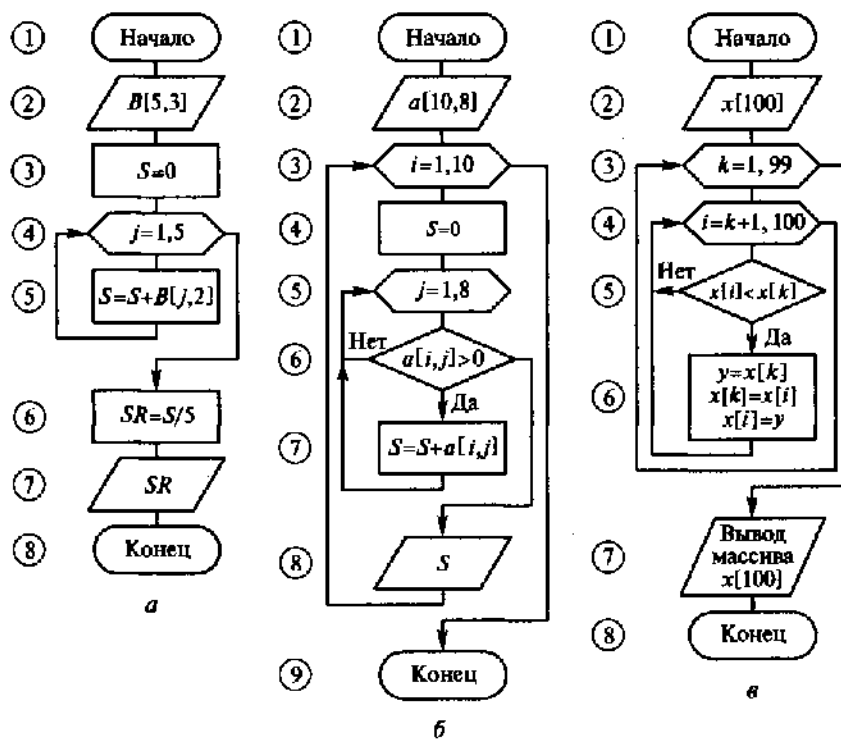


Рис. 7.9. Алгоритмы с использованием массивов:

a — алгоритм решения задачи, представленной в примере 1; *б* — алгоритм решения задачи, представленной в примере 2; *в* — алгоритм решения задачи, представленной в примере 3.

$V[1,1] = 3$ (оценка Андреева по математике); $V[2,3] = 3$ (оценка Иванова по иностранному языку); $V[4,3] = 4$ (оценка Сорокиной по информатике). Обратите внимание, что фамилии студентов не являются элементами массива.

Пример 1. Найдем средний балл студентов по информатике. В блоке 2 (рис. 7.9, а) производится ввод значений элементов массива, размерность которого 5×3 . В программе эта операция может быть реализована разными способами: из файла или с клавиатуры. Далее в цикле (блок 5) происходит суммирование элементов 2-го столбца в переменной S , так как предмету «Информатика» соответствует 2-й столбец в таблице. Подсчет среднего значения, которое обозначено переменной SR , производится в блоке 6 делением суммы всех оценок по информатике (S) на число студентов (5).

В цикл, называемый *внешним*, могут входить один или несколько вложенных циклов, называемых *внутренними*. Организации как внешнего, так и внутреннего циклов осуществляются по тем же правилам, что и организация простого цикла. Параметры внешнего и внутреннего циклов разные и изменяются не одновременно, при одном значении параметра внешнего цикла параметр внутреннего цикла принимает поочередно все значения от начального до конечного, т.е. внутренний и внешний циклы не могут пересекаться, а должны быть вложенными.

Пример 2. Вычислить сумму положительных элементов каждой строки матрицы A (10×8).

Предполагается, что все значения элементов матрицы A заданы (блок 2) (рис. 7.9, б). Для вычисления суммы положительных элементов строки матрицы необходимо организовать цикл, перебирающий все элементы одной строки. Параметром этого цикла (блоки 5—7) будет номер столбца j . Перед циклом задается начальное значение суммы $S = 0$ (блок 4). После окончания цикла результат выводится на печать (блок 8). Если все эти действия повторить во внешнем цикле, изменяя индекс строки i (блок 3), то будут вычислены все 10 сумм.

В рассмотренном примере внешний цикл обязательно должен быть по i (индексу строки), так как в противном случае, если i и j поменять местами, были бы вычислены суммы положительных элементов каждого столбца. Часто встречаются задачи, для решения которых не важно, по какому параметру организуются внешний и внутренний циклы.

Пример 3. Упорядочить элементы массива $X(100)$, расположив их по возрастанию в том же массиве.

Для решения данной задачи используется метод, суть которого заключается в следующем. Организуется внутренний цикл. Первый элемент массива или его части x_k сравнивается со всеми последующими элементами x_i . Всякий раз, когда элемент x_k будет больше какого-либо последующего элемента x_i , необходимо осуществить перестановку x_k и x_i . После окончания цикла по i значение x_k будет наименьшим из рассматриваемых значений элементов массива, т.е. при первом проходе цикла на первом месте будет стоять самое маленькое значение, а остальные элементы не будут упорядочены. Для упорядочивания остальных элементов указанные действия необходимо повторить, проведя все описанные выше операции для второго, третьего и следующих элементов, т.е. организовав внешний цикл, в котором k изменяется от 1 до 99, а во внутреннем цикле (i) элементы перебираются от $(k + 1)$ -го до 100-го. Когда останется последний элемент, процесс упорядочивания следует прекратить.

Схема алгоритма решения представлена на рис. 7.9, в. Блок 5 проверяет выполнение условия $x_i < x_k$, а блок 6 осуществляет перестановку элементов массива, используя промежуточную переменную y .

7.2.5. Вспомогательные алгоритмы

Эффективным методом построения алгоритмов является метод пошаговой детализации. При этом сложная задача разбивается на ряд более простых подзадач. Для каждой подзадачи составляется свой вспомогательный алгоритм. Метод пошаговой детализации универсален.

Вспомогательный алгоритм (процедура) — это алгоритм, который описывает решение какой-либо подзадачи и вызывается по имени из основного алгоритма.

Вспомогательные алгоритмы создаются и тогда, когда есть необходимость многократного использования одного и того же набора действий в одном или разных алгоритмах. Вспомогательные алгоритмы должны быть состыкованы между собой в процессе «сборки» основного алгоритма. Для этого используют заголовок вспомогательного алгоритма; с его помощью этот алгоритм вызывают (обращаются к его работе) из других вспомогательных или основного алгоритмов. При составлении и использовании вспомогательных алгоритмов важно знать, что является для них исходными данными (аргументами), а что результатами.

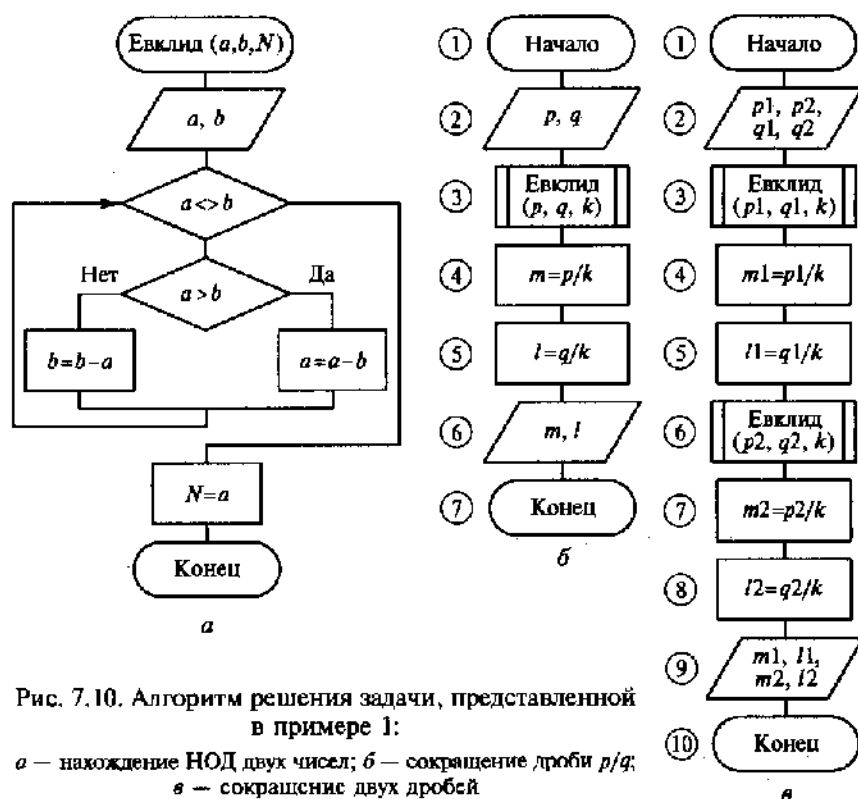
Записывая программу для компьютера на языке программирования, вспомогательные алгоритмы можно организовать, например, как подпрограммы. Правила обращения к ним и возврата из

них в основную программу определяются конкретным языком программирования. Подпрограммы общего назначения могут объединяться в библиотеки подпрограмм, а иногда образовывать набор стандартных функций. Например, функции Max, Min и другие являются вспомогательными алгоритмами.

Пример 1. Рассмотрим решение задачи сокращения дроби. Исходными данными являются числитель p и знаменатель q — целые положительные числа. Вспомогательным алгоритмом будет алгоритм Евклида для нахождения наибольшего общего делителя (НОД) двух чисел (рис. 7.10, а).

Механизм выполнения алгоритма Евклида (нахождение наибольшего общего делителя N целых чисел a и b) следующий. Задаются два числа: a и b . Если они равны, то число a берется в качестве ответа N и работа алгоритма завершается, а в противном случае определяется большее из чисел и оно заменяется разностью большего и меньшего числа до тех пор, пока числа не сравняются.

Основной алгоритм (рис. 7.10, б) основан на вызове вспомогательного алгоритма (блок 3), который вставляется в основной



алгоритм вместо блока 3, но с подстановкой значений p , q и k вместо a , b и N . Такой механизм называется механизмом передачи параметров. Вспомогательный алгоритм, представленный на рис. 7.10, a , построен для любых произвольных чисел a и b и получением НОД в переменной N . При его вызове в блоке 3 (см. рис. 7.10, b) вместо абстрактных переменных a и b подставляются конкретные значения p и q , а результат будет передан в переменную k . Связь переменных осуществляется через заголовок вспомогательного алгоритма. В блоках 4—5 выполняется сокращение дроби и получаются новые числитель (m) и знаменатель (l).

Усложним задачу, сократим две дроби с числителями p_1 , p_2 и знаменателями q_1 , q_2 соответственно. Алгоритм представлен на рис. 7.10, b , вспомогательный алгоритм вызывается дважды в блоке 3 для первой дроби, в блоке 6 — для второй дроби. Попробуйте объяснить сами, почему достаточно одной переменной k и имеет ли она одинаковые значения при сокращении первой и второй дробей.

Контрольные вопросы

1. Объясните смысл операции присваивания.
2. Запишите логическое выражение, истинное при выполнении условия: x лежит вне отрезков $[a, b]$ и $[c, d]$.
3. Что такое тело цикла?
4. Может ли цикл с предусловием не выполниться ни разу?
5. Что такое вспомогательный алгоритм?
6. Может ли вспомогательный алгоритм вызываться в цикле?
7. Приведите примеры задач, требующих использования массива.
8. Вычислите периметр и площадь прямоугольного треугольника, если известны катет и острый угол.
9. Цены на два вида товаров возросли на p процентов. Выведите старые и новые цены.
10. Составьте алгоритм нахождения суммы кубов целых чисел в интервале от n_1 до n_2 .
11. Определите число положительных чисел среди заданных a , b , c .
12. Получите значения функции $y = \sin(x) + x$ для интервала $[-2, 2]$.
13. В забеге участвовали шесть спринтеров. Найдите лучшее время.
14. Может ли цикл со счетчиком быть бесконечным?
15. Приведите два примера, требующих использования одного вспомогательного алгоритма.
16. Вычислите произведение и число всех значений функции $y = \sin \frac{2x}{\pi - 4,1} + \cos^3 2x$, для которых $y > 0,4$ или $y < -3$.
17. В целочисленном массиве второй отрицательный элемент замените минимальным.
18. В целочисленном массиве предпоследний отрицательный элемент замените максимальным.

Глава 8



ПРОГРАММИРОВАНИЕ НА VISUAL BASIC 6

Создание программных продуктов — трудоемкий процесс, основанный на определенной технологии и инструментарии его разработки. Составленный алгоритм решения задачи следует перевести на язык, понятный ЭВМ. Такой язык называется языком программирования, запись алгоритма на таком языке называется программой, а процесс перевода алгоритма на указанный язык — программированием. В данной главе изложены основы программирования на языке Visual Basic.

Объектно-ориентированное и визуальное программирование в настоящее время — наиболее популярная технология программирования. Эта технология появилась в результате развития технологии нисходящего структурного программирования.

Для современных программ обязательным элементом является красочное оформление — наличие эффектных заставок, картинок-вставок, удобного диалогового интерфейса. Язык Visual Basic включает в себя язык программирования и язык оформления программы. Поэтому даже начинающий программист может оформить свою программу таким образом, что она может соперничать (по внешнему виду) с программой фирмы Microsoft.

8.1. ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Программирование — это написание текста на алгоритмическом языке. Целью программирования является описание процессов обработки данных.

Система программирования — это язык и средства разработки прикладных программ. Современное инструментальное средство для разработки программ представляет собой интегрированную среду, которая строится на интерактивном принципе, обеспечивающем повышение эффективности программирования и полный набор действий для разработки программы и анализа результатов ее выполнения.

В состав современной интегрированной среды разработки программ входят: текстовый редактор, транслятор, редактор связей, библиотеки подпрограмм, программа-отладчик, развитая система помощи и подсказок.

Существующие системы программирования подразделяются на компилирующие и интерпретирующие, в соответствии с типом их трансляторов.

Трансляторы — это программы, которые переводят текст с языка программирования на машинный.

Они могут быть двух типов: компиляторы и интерпретаторы.

Компилятор — это транслятор, в котором трансляция и выполнение программы разнесены во времени: сначала весь текст транслируется без выполнения программы, а затем полученная программа выполняется (Паскаль, Си).

Интерпретаторы — это трансляторы, в которых текст транслируется и выполняется пооператорно, т.е. трансляция и выполнение программы совмещены во времени (интерпретатор команд ОС).

Основные этапы разработки программы в интегрированной среде: написание текста исходных данных и программы, трансляция программы, редактирование связей программы с подпрограммами, выполнение программы, анализ полученных результатов.

Наиболее общей классификацией языков программирования является классификация по степени зависимости от машинного языка (рис. 8.1).

Машинный язык — это система команд компьютера. Программы, написанные на машинном языке, не требуют компиляции. Машинно-ориентированные языки — это мнемокоды, автокоды, языки ассемблера. В них коды операции и адреса операндов представляются идентификаторами и числами, т.е. используются символические адреса и мнемокоды. На ассемблере можно писать программы более эффективные, чем программы, полученные в результате трансляции с языка высокого уровня.

Проблемные языки предназначены для пользователей, не умеющих программировать, и направлены на решение узкого круга задач. На них определяется, что делать, а не как это делать, в терминах, характерных для данной проблемной области. Для ра-



Рис. 8.1. Классификация языков программирования

боты с проблемно-ориентированным языком должны быть разработаны программы для анализа запросов пользователя, контроля допустимости вводимых данных и для решения поставленной задачи.

Процедурные языки — это машинно-независимые языки для описания алгоритмов решения задачи. С их помощью описывается процесс обработки данных в терминах языка. Тексты на этих языках определяют, как реализуется алгоритм обработки данных в виде процедур обработки данных. Такие языки применяют специалисты, знающие математическую модель решения задачи, методы ее решения и программирование.

Объектно-ориентированные языки основаны на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств.

На практике языки программирования не являются чисто процедурными, проблемными или объектно-ориентированными, они содержат в себе черты различных языков, поэтому зачастую на процедурном языке можно разработать объектно-ориентированную программу.

На протяжении всех лет существования практика программирования требовала совершенствования технологических приемов и создания на их основе таких средств программирования, которые упростили бы процесс разработки программ, позволяя создавать все более сложные программные системы. Первые программы были организованы очень просто. Они состояли из собственно программы на машинном языке и обрабатываемых данных. Сложность программ ограничивалась способностью программиста одновременно мысленно отслеживать последовательность выполняемых операций и местонахождение большого числа данных. Создание сначала ассемблеров, а потом и языков высокого уровня сделало программу более обозримой и, естественно, позволило увеличить ее сложность. Появление в языках средств, позволяющих оперировать подпрограммами, существенно снизило трудоемкость разработки программ.

Подпрограммы — это программы небольшого объема, к которым возможно многократное обращение в программе. Подпрограммы можно было сохранять и использовать в других программах. В результате были накоплены огромные библиотеки расчетных и служебных программ, которые по мере надобности вызывались из разрабатываемой программы.

Типичная программа того времени состояла из основной программы, области глобальных данных и набора подпрограмм (см. подразд. 7.2), выполняющих обработку всех данных или их части.

Недостатком такой технологии было то, что при увеличении числа подпрограмм возрастала вероятность искажения части гло-



Рис. 8.2. Структура программы:

$1, 2, \dots, N$ — подпрограммы

бальных данных какой-либо подпрограммой (рис. 8.2). Необходимость исключения таких ошибок привела к идее использования в подпрограммах локальных данных. И вновь сложность разрабатываемой программы стала ограничиваться возможностью программиста отслеживать процессы обработки данных уже на новом уровне. К этому добавились проблемы согласования интерфейса при ведении разработки несколькими программистами.

В настоящее время многие среды программирования используют элементы интерфейса, близкие к интерфейсу Windows: коман-

дные кнопки, флажки, окна сообщений и диалога и т.д. В результате встал вопрос создания технологии разработки сложных программных продуктов, снижающей вероятность появления ошибок. Усилиями многих программистов такая технология была создана и получила название «структурное программирование».

В основе структурного программирования лежит разбиение на части сложных систем с целью последующей реализации в виде отдельных небольших подпрограмм. Поддержка принципов структурного программирования была заложена в основу так называемых процедурных языков программирования. Наиболее известными языками этой группы являются PL/I, Pascal, Си.

Рост сложности и размеров разрабатываемого программного обеспечения потребовал развития структурирования данных и, соответственно, в языках появилась возможность определения пользовательских типов данных. Одновременно усилилось стремление разграничить доступ к глобальным данным программы для уменьшения числа ошибок. Результатом стало появление и развитие технологии модульного программирования.

Модульное программирование предполагает выделение групп подпрограмм, использующих одни и те же глобальные переменные, в отдельные модули (библиотеки подпрограмм). Связи между модулями осуществляются через специальный интерфейс. Эту технологию поддерживают современные версии языков Pascal и Си (C++), языки Ada и Modula. Использование модульного программирования существенно упрощает ведение разработки программ несколькими программистами, каждый из которых разрабатывает свои модули.

Практика программирования показывает, что структурный подход в сочетании с модульным программированием позволяет получать достаточно надежные программы, размер которых не превышает 100 000 операторов.

Под *надежностью программы* понимается вероятность безотказной ее работы в течение определенного промежутка времени.

Стремление уменьшить число связей между отдельными частями программ привело к появлению объектно-ориентированного программирования (ООП). По своей сути ООП — это создание программ из объектов, подобно тому, как их блоков и различных деталей строятся дома. Одни объекты приходится полностью создавать самостоятельно, тогда как другие можно позаимствовать в готовом виде из разнообразных библиотек.

В последнее время стала развиваться еще одна технология программирования — case-технология.

Case-технология — программный комплекс, автоматизирующий весь технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем. Основное достоинство case-технологии — поддержка коллективной работы над проектом за счет возможности работы в локальной сети разработчиков, экспорта/импорта любых фрагментов проекта, организационного управления проектом.

Контрольные вопросы

1. Какие технологии программирования являются наиболее востребованными в настоящее время?
2. Что представляет собой технология структурного программирования?
3. Что такое транслятор?
4. Назовите основное отличие компилятора от интерпретатора.
5. Зачем человечеству необходима технология программирования?

8.2. КОНЦЕПЦИЯ ПРОГРАММИРОВАНИЯ НА VISUAL BASIC

8.2.1. Структура объектов

В процессе моделирования предметная область представляется в виде совокупности объектов.

Изменение свойств объекта не происходит само по себе, для их изменения необходимо выполнить определенные операции (действия). Изменению каждого свойства обычно соответствует определенная операция. Так, для превращения неотформатированной дискеты в форматированную необходимо провести операцию *Форматирование*.

Для выполнения операции по изменению значения свойства объекта обычно требуется метод.

Объект — реально существующий предмет в окружающем нас мире, обладающий состоянием, поведением и индивидуальнос-

тью. Состояние объекта определяется перечнем всех возможных свойств и текущим значением каждого из них. Перечень свойств объекта обычно статический, а их значения меняются. Другими словами, для определения объекта надо задать значения его свойств. Объекты не существуют изолированно, они являются элементами систем, подвергаются воздействию других объектов и сами воздействуют на другие объекты. В процессе функционирования сложных систем входящие в них объекты постоянно обмениваются информацией.

Воздействие одного объекта на другой называется *сообщением*. Таким образом, функционирование объектно-ориентированного программного обеспечения (точнее, программного обеспечения, построенного по объектно-ориентированной технологии) заключается в передаче сообщений между объектами и реакциях объектов на полученные сообщения.

При использовании технологии ООП решение представляется в виде результата взаимодействия отдельных функциональных элементов некоторой системы, имитирующей процессы, происходящие в предметной области поставленной задачи.

Функциональные элементы системы, параметры и поведение которой определяются условием задачи, обладающей самостоятельным поведением (т.е. умеющие выполнять некоторые действия, зависящие от полученных сообщений и состояния элемента), получили название *объектов*. Основной единицей в объектно-ориентированном программировании является программный объект, который объединяет в себе как описывающие его данные (свойства), так и средства обработки этих свойств (методы).

Классы объектов являются шаблонами, определяющими наборы свойств, методов и событий. По этим шаблонам создаются объекты (например, в языке Visual Basic основными являются классы объектов, реализующие графический интерфейс приложения).

Свойство — характеристика объекта, его параметр. Все объекты наделены определенными свойствами, которые в совокупности выделяют объект из множества других объектов.

Например, объект можно представить перечислением присущих ему свойств:

ОБЪЕКТ_А (свойство-1, свойство-2, ..., свойство-*k*).

Свойства объектов различных классов могут «пересекаться», т.е. возможны объекты, обладающие одинаковыми свойствами:

ОБЪЕКТ_В (...свойство-*n*, свойство-*m*, ..., свойство-*r*,...),

ОБЪЕКТ_С (...свойство-*n*, ..., свойство-*r*,...).

Одним из свойств объекта является метод его обработки.

Метод — программа действий над объектом или его свойствами.

Метод рассматривается как программный код, связанный с определенным объектом и осуществляет преобразование свойств, изменяет поведение объекта.

Объект может обладать набором заранее определенных встроенных методов обработки либо созданных пользователем или заимствованных в стандартных библиотеках, которые выполняются при наступлении заранее определенных событий (например, однократное нажатие левой кнопки мыши, вход в поле ввода, выход из поля ввода, нажатие определенной клавиши и т. п.).

С развитием систем обработки данных создаются стандартные библиотеки методов, в состав которых включаются типизированные методы обработки объектов определенного класса (аналог — стандартные подпрограммы обработки данных при структурном подходе), которые можно заимствовать для различных объектов.

Событие — изменение состояния объекта.

Внешние события генерируются пользователем (например, клавиатурный ввод или нажатие клавиши мыши, выбор пункта меню), а внутренние события генерируются системой.

Программный продукт, созданный с помощью инструментальных средств объектно-ориентированного программирования, содержит объекты с их характерными свойствами, для которых разработан графический интерфейс пользователя. Как правило, работа с программным продуктом осуществляется с помощью экранной формы. На экранной форме размещаются объекты управления, которые содержат методы обработки, вызываемые при наступлении определенных событий. Экранные формы также используются для выполнения заданий и перехода от одного компонента программного продукта к другому. Каждый объект управления обладает определенными свойствами, значения которых могут изменяться. Для объектов управления уточняется перечень событий и создаются пользовательские методы обработки — программный код на языке программирования в виде событийных процедур. Объект, создающийся по шаблону класса объектов, является экземпляром класса и наследует весь набор свойств, методов и событий этого класса.

8.2.2. Объектно-ориентированный язык программирования Visual Basic

Visual Basic (VB) использует визуальный метод создания графического интерфейса и объектный метод построения его программного кода.

Событие — это действие, которое распознает объект. Для каждого события можно запрограммировать реакцию объекта на произошедшее событие.

Реакция — это последовательность действий; обычно ее называют событийной процедурой. В событийной процедуре могут участвовать несколько объектов.

Графический интерфейс образуют управляющие элементы (командные кнопки, переключатели, радиокнопки, текстовые поля и др.) и формы.

Форма — это объект, представляющий собой окно на экране, в котором размещаются управляющие элементы.

В Visual Basic экранная форма хранится в отдельном файле, имя которого имеет расширение .frm.

Разрабатываемое на языке VB приложение называется *проектом*. Проект включает в себя форму с размещенными на ней управляющими элементами и программные модули событийных процедур, которые описывают поведение объектов приложения и взаимодействие объектов между собой.

Программный модуль — это хранящийся в отдельном файле программный код, содержащий тексты событийных процедур. Имя этого файла имеет расширение .bas.

Под термином «программный модуль» будем понимать программный код, который относится к одной экранной форме (программный модуль данной экранной формы). Программный модуль может быть маленьким — содержать всего несколько строк программного кода, а может быть очень большим — содержать сотни и тысячи строк кода.

Отметим следующую особенность: если экранная форма со своими объектами в работающем приложении видна пользователю и он ими управляет, то программный код этой формы пользователю не виден. Код виден только программисту, который его пишет, создавая (проектируя) приложение.

Visual Basic хранит каждый проект в отдельном файле, имя которого имеет расширение .vbp.

Программный код проекта существует не сам по себе, он не оторван от экранной формы, а привязан к отдельным ее элементам (объектам). Более того, часть кода, которая относится только к одному объекту, в свою очередь, может состоять из нескольких фрагментов — процедур. Большинство процедур, из которых состоит программный код Visual Basic, привязаны к событиям и называются событиями-процедурами.

Процесс создания проекта в VB состоит из пяти этапов.

1. Постановка задачи — составление по возможности точного и понятного словесного описания того, как должно работать будущее приложение, что должен делать пользователь в процессе его работы. Это описание должно объяснять и то, как будет выглядеть экранная форма (окно) этого приложения, в каком виде будет представлена информация, которую нужно преобразовать с помощью этого приложения (например, исходные данные и результаты).

2. Разработка интерфейса — создание экранной формы (окна приложения) со всеми находящимися на этой форме объектами и свойствами этих объектов.

3. Программирование — определение того, какие события будут происходить в процессе работы приложения, составление алгоритмов процедур для этих событий и написание программы (программных кодов этих процедур).

4. Отладка программы — устранение логических ошибок в процедурах и достижение того, чтобы приложение удовлетворительно работало в среде проектирования.

5. Сохранение проекта и, при желании, компиляция — превращение проекта в исполняемое приложение, способное работать самостоятельно — за пределами среды проектирования.

Рассмотрим пример создания Windows-приложения в среде Visual Basic для задачи, которая традиционно является первой задачей в любой среде программирования и называется «Приветствие».

Постановка задачи. Вывести на экран окно с кнопкой *Старт*, после нажатия на которую левой клавишей мыши появится второе окно с сообщением «Здравствуй, пользователь!».

Разработка интерфейса. На этом этапе создается экранная форма. Для ее создания необходимо познакомиться со средой Visual Basic.

Система проектирования (программирования) Windows-приложений Visual Basic 6 — это тоже Windows-приложение. Но это приложение значительно сложнее тех проектов, которые вы будете создавать с его помощью.

Вход в систему можно выполнить одним из следующих способов.

1. Если пиктограмма Visual Basic 6 находится на вашем рабочем столе, то достаточно дважды щелкнуть ее левой клавишей мыши.

2. В противном случае в меню Windows *Пуск* можно открыть подменю *Программы*, в нем выбрать группу программ «Microsoft Visual Basic 6.0», а в ней — пункт (программу) «Microsoft Visual Basic 6.0».

3. Можно также с помощью *Проводника* Windows найти папку с файлом Vb6.exe и запустить его.

В любом из перечисленных случаев на экране вашего монитора должна появиться *Главная панель* проекта Visual Basic 6, поверх которой будет находиться окно *New Project (Новый проект)* (рис. 8.3, а). В окне три закладки (страницы): *New*, *Existing* и *Recent*. Рассмотрим использование этих закладок по порядку.

Закладка *New* содержит объекты для создания нового проекта.

Использование закладки *Existing* позволяет просмотреть все множество файлов с проектами Visual Basic, которые находятся в текущем каталоге (папке), и выбрать интересующий вас файл. Кроме того, вы имеете возможность просмотреть и другие папки вашего компьютера.

Использование закладки *Recent* позволяет увидеть прежде всего те файлы проектов, над которыми шла работа недавно. Первым

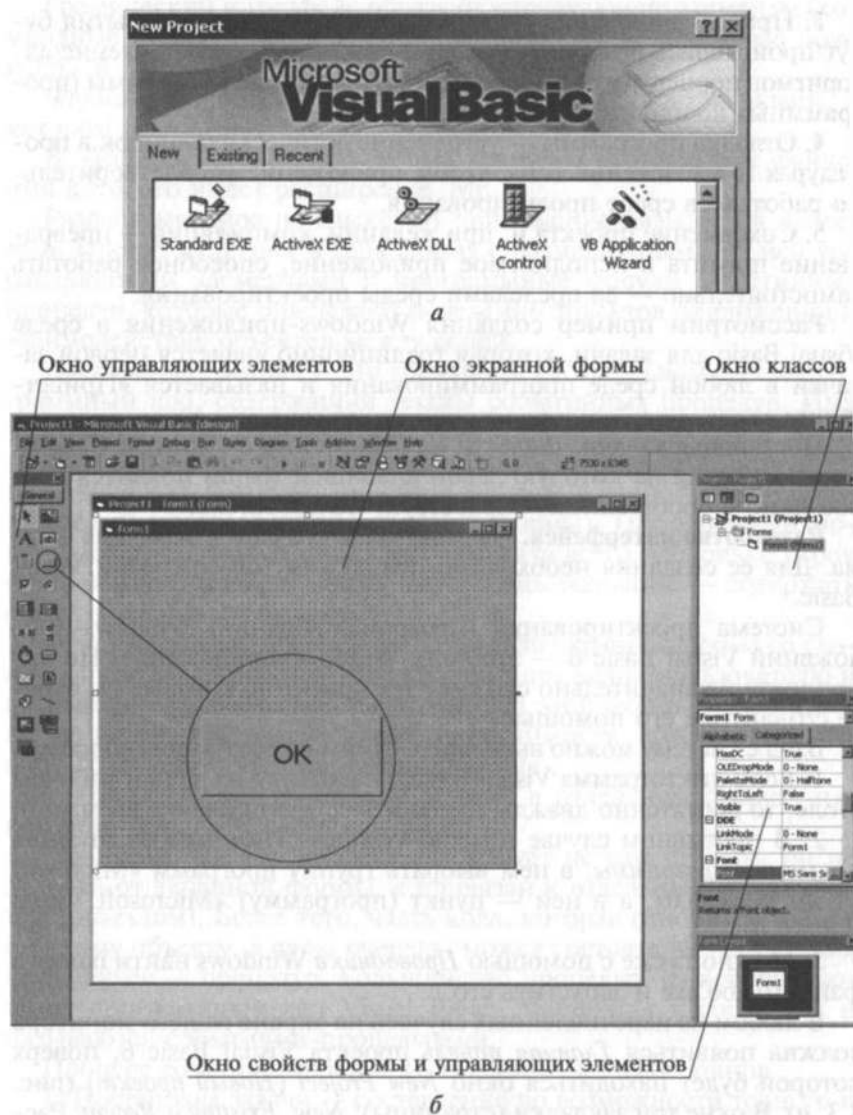


Рис. 8.3. Создание Проекта VB:

a — запуск системы проектирования; *б* — Главная панель проекта

в списке стоит файл проекта, над которым вы (или кто-то другой на вашем компьютере) работали перед данным сеансом работы. Второй в списке — предшествующий ему проект и т.д. Это очень удобно, так как работа над одним проектом иногда затягивается на несколько дней.

После выбора закладки *New* необходимо открыть объект *Standard EXE*. На экране появится *Главная панель* проекта. Она предназначена для разработки нового проекта в среде Visual Basic 6 (рис. 8.3, б).

Рассмотрим элементы главного окна системы проектирования Visual Basic. После того как на экране монитора появится *Главная панель* проекта, на ней следует открыть окна, которые понадобятся вам на начальной стадии проектирования — при создании экранной формы вашего приложения: окно инструментов, окно экранной формы и окно свойств этой формы. Эти, а также и некоторые другие окна могут появиться на главной панели по умолчанию, но вы можете оставить только те, которые нужны вам в данный момент, а остальные можно убрать, щелкнув по кнопке  (*Закрытие окна*). Окно экранной формы расположено в центре главной панели и имеет заголовок *Project1 — Form1 (Form)*. По умолчанию в этом окне уже есть заготовка экранной формы. Она имеет заголовок *Form1*. Экранная форма является первым объектом, с которым вы начинаете работать. Другими объектами будут элементы будущего приложения, которые вы разместите на этой форме.

В языке Visual Basic рассматривается такое понятие как «класс объектов *Form*». Это совокупность общих характеристик любой экранной формы. Когда создается конкретная экранная форма или какой-нибудь иной конкретный объект, можно использовать уже имеющийся в системе список свойств класса этих объектов. Именно этот список вы и видите в правой части рис. 8.3, б — в окне *Properties* (окне свойств). С помощью окон экранной формы и свойств этой формы можно устанавливать различные свойства формы на начальной стадии проектирования интерфейса.

Прежде всего, установите значения ее размеров — свойство *Width* (*Ширина*) и *Height* (*Высота*). Можно, конечно, установить эти значения с помощью окна *Properties*, внося конкретные числа в правую колонку таблицы, находящейся в этом окне. Но лучше это сделать «на глазок», растягивая мышью стороны экранной формы. При этом значения в окне *Properties* будут изменяться автоматически.

Значения свойств положения формы на экране — свойств *Left* (*Левый край*) и *Top* (*Верхний край*) — легко установить тоже «на глазок» или с помощью окна *Form Layout Window* (окна расположения формы). Открыть это окно можно, выбрав команду  *Form Layout Window* меню *View*. А вот для установки значений имени (идентификатора) формы, надписи в строке заголовка формы, цвета фона формы окно *Properties* будет необходимо. Этими свойствами соответственно являются *Name*, *Caption*, *BackColor*. Значения этих свойств набираются с помощью клавиатуры, а значение последнего свойства *BackColor* выбирается с помощью раскрыва-

ющейся панели с двумя закладками после щелчка по кнопке  в строке «BackColor» окна *Properties*.

Рассмотрим несколько объектов, которые в дальнейшем будем использовать в наших проектах. Для их создания вам понадобится окно *Toolbox* (окно инструментов). Если этого окна еще нет на *Главной панели*, то следует его открыть, например, выбрав команду  меню *View*.

В окне *Toolbox* находятся пиктограммы (инструменты), с помощью которых можно создавать любое число объектов управления одного и того же класса. В частности, с помощью инструмента *Label* можно создать любое число объектов «Метка». Объект «Метка» используется для вывода надписей, заголовков и любого небольшого фрагмента текста.

 — пиктограмма этого инструмента.

Рассмотрим технологию создания меток на примере размещения на экранной форме одной из них. Щелкните по пиктограмме «Label» в окне *Toolbox*. Поместите указатель мыши в то место экранной формы, где будет находиться левый верхний угол будущего объекта. При нажатой левой клавише мыши «проташите» указатель в то место, где будет находиться правый нижний угол будущего объекта. Этот процесс будет сопровождаться растяжением пунктирной рамки — границы создаваемого объекта. Размеры и положение появившегося прямоугольника можно подкорректировать.

Сделайте это следующим образом:

- «ухватив» какой-либо из восьми маркеров (квадратиков, которыми окружен прямоугольник) мышью, вы можете изменить размеры объекта;
- установив курсор внутри прямоугольника, с помощью мыши вы можете переместить объект в пределах экранной формы, не меняя его размеров.

Основными свойствами объекта «Метка» являются:

- идентификатор объекта (Name);
- надпись, в которой помещается текст (Caption);
- тип, размер и цвет шрифта (Font, ForeColor);
- цвет фона (BackColor).

Если объект «Метка» используется для вывода текстовой информации, то объект «Текстовое поле» используется как для ее ввода, так и для вывода.

В окне *Toolbox* найдите пиктограмму инструмента, с помощью которого можно создать любое число объектов управления класса «Текстовое поле». Это инструмент *TextBox*.

 — пиктограмма этого инструмента.

У текстового поля много тех же свойств, что и у метки. Посмотрите, какие свойства вам уже знакомы? У текстового поля

нет свойства *Caption*. Вместо него есть свойство *Text*. Используя это свойство, можно вводить очень большой объем информации.

Перейдем теперь к объекту «Командная кнопка».

 — пиктограмма инструмента CommandButton.

Главная его характеристика — это не какое-нибудь свойство, а событие. Оно заключается в щелчке мышью по этой кнопке.

Программирование. Для написания программного кода и привязки его к определенному событию необходимо раскрыть окно *Code* (окно программного кода) (если его еще нет на *Главной панели*). Сделать это можно, выбрав команду  *Code* меню *View*. В верхней части этого окна есть две строки раскрывающихся (но пока не раскрытых) списков: списка уже созданных объектов (левая строка) и списка еще не существующих процедур (правая строка).

Процедура — это фрагмент программного кода, с помощью которого решается какая-то локальная задача. Часто (но не всегда) процедура называется событием. Поэтому правый список в окне программного кода, хотя и называется списком процедур, лучше было бы назвать списком возможных событий для выбранного объекта.

Раскрыв «Список объектов», вы получите возможность увидеть идентификаторы всех созданных вами объектов на экранной форме, а также указатель на саму экранную форму (*Form*).

Найдите в этом списке идентификатор того объекта, для которого вы собираетесь создать процедуру. Для нашего примера это «КоманднаяКнопка». Щелкните по ней мышью. Теперь раскроется правый список (список процедур), который будет содержать все возможные события для этого объекта. А в самом окне программного кода появится при этом заготовка для вашего программного кода — первая и последняя строки процедуры для самого распространяемого события *Click*, относящегося к объекту «Командная кнопка». Процедура называется «КоманднаяКнопка_Click».

Первая строка программы начинается со слов «Private Sub», а заканчивается программа словами «End Sub» — это служебные слова языка Бейсик. Visual Basic в окне программного кода выделяет эти слова синим цветом. Между указанными двумя строками заготовки необходимо поместить строки программного кода.

Для рассматриваемого примера экранная форма будет содержать единственный объект — командную кнопку *Старт*. Свойствами этого объекта являются имя кнопки, размеры кнопки, ее положение в окне, ее цвет, надпись на кнопке и многое другое.

Значение свойства *цвет* — «серый», значение свойства *надпись* — «Старт», значение свойства *имя* — «КоманднаяКнопка» и т.д.

В работающей программе, после того как пользователь щелкнет мышью по кнопке *Старт*, должно появиться окно сообще-

ния (рис. 8.4). В этом окне уже не один, а три объекта: кнопка *OK*, надпись «Здравствуй, пользователь!» и графический элемент .

Для приложения, приведенного в примере, программный код очень небольшой и состоит из единственной процедуры-события:

```
Private Sub КоманднаяКнопка_Click()  
MsgBox «Здравствуй, пользователь!», 64, «Приветствие»  
End Sub
```

С функцией *MsgBox* мы познакомимся позже.

После того как текст процедуры введен вами между двумя строками заготовки и система Visual Basic молча восприняла этот текст (не выдала вам никаких предупреждений о синтаксических ошибках), вы можете запустить вашу программу.

Запускать программу можно разными способами:

- с помощью опции *Run* и команды *Start* главной панели проекта;
- с помощью кнопки  (*Start*) линейки инструментов той же панели;
- с помощью клавиши [F5] клавиатуры.

Завершить работу программы (закрыть окно приложения) можно тоже по-разному, например:

- с помощью кнопки  (*End*) на линейке инструментов;
- с помощью стандартного элемента Windows — системной кнопки закрытия окна в правом верхнем углу окна приложения.

Отладка программы. Попытка запустить программу не всегда приводит к успеху. Очень часто (особенно, когда программа не такая простая, как в рассмотренном примере) попытка запуска приводит к появлению «грозных» сообщений системы Visual Basic об ошибках. При этом не нужно отчаиваться и ругать себя — не ошибается только тот, кто ничего не делает. Ошибки можно и нужно исправить. Для этого Visual Basic предоставляет вам разнообразные средства отладки, представленные на панели *Debug* (табл. 8.1).



Рис. 8.4. Работа приложения «Приветствие»

Таблица 8.1

Элементы панели инструментов *Debug*

Кнопка	Название	Назначение
	<i>Start</i>	Запуск приложения
	<i>Break</i>	Временная остановка выполнения программы (пауза)
	<i>End</i>	Завершение работы приложения и переход к этапу разработки
	<i>Toggle Breakpoint</i>	Установка контрольной точки, т.е. определение шага в программе, перед которым выполнение приложения временно прекращается (при этом программа переводится в так называемый режим прерывания)
	<i>Step Into</i>	Программа находится в режиме прерывания, и выполняется один последующий оператор, причем вызываемые процедуры в этом случае будут выполняться пошагово (режим <i>Step Into</i>)
	<i>Step Over</i>	Выполнение одного последующего оператора в режиме прерывания, причем выполнение процедуры будет осуществляться за один шаг
	<i>Step Out</i>	Автоматическое завершение выполняемой процедуры в режиме отладки <i>Step Into</i>
	<i>Locals Window</i>	Открытие окна <i>Locals</i> , содержащего, в свою очередь, окна <i>Call Stack</i> и <i>Watch</i> , которые позволяют отслеживать текущие значения различных переменных
	<i>Immediate Window</i>	Открытие окна <i>Immediate</i> для выполнения проверочного кода
	<i>Watch Window</i>	Отслеживание значений переменных при помощи окна <i>Watch</i>
	<i>Quick Watch</i>	Отображение значения одной переменной или выражения в окне <i>Quick Watch</i>
	<i>Call Stack</i>	Контроль вызова процедур и функций с помощью окна <i>Call Stack</i>

Можно сохранить приложение (как экранную форму, так и весь проект в целом) для доработки или модернизации в последующих сеансах работы на компьютере.

В окне проводника проекта щелкните по имени вашего проекта, которое задано по умолчанию (Project1) и измените его. В раскрывшемся окне *Properties* установите значение единственного

свойства (Name). А теперь воспользуйтесь кнопкой *Сохранить* инструментальной линейки главной панели проекта. Щелчок по ней вызовет появление окна диалога системы Windows. Нажатие кнопки *Сохранить* вызывает сохранение файла формы и появление такого же окна для сохранения файла проекта.

После того как вы и в этом окне нажмете кнопку *Сохранить*, ваше приложение будет сохранено для дальнейшей модернизации или доработки.

Создание исполняемого файла приложения — exe файла. Итак, вы создали ваш первый работающий проект в системе Visual Basic — запустили программу и убедились в том, что она работает так, как вы задумали. Но считать ваш проект законченным Windows-приложением еще рано, так как он может работать только в среде проектирования при раскрытой главной панели проекта Visual Basic.

Для того чтобы Windows-приложение могло работать независимо от среды проектирования (автономно), нужно произвести преобразование (трансляцию) проекта в исполняемый exe-файл. Для трансляции проекта в исполняемый exe-файл используется опция *File* главной панели проекта и в ней — команда *Make ИмяФайла.exe* (ИмяФайла — это имя вашего проекта). После щелчка мышью этой команды появится стандартное диалоговое окно *Windows*, с помощью которого Visual Basic попросит у вас подтверждения того, что exe-файл будет иметь имя, совпадающее с именем файла вашего проекта, или другое указанное вами имя.

Контрольные вопросы

1. Что такое объект?
2. Как можно изменять свойства объектов?
3. Почему ООП называют событийным программированием?
4. Что такое событие?

8.3. ВЫРАЖЕНИЯ И ФУНКЦИИ

Многие программы выполняют различные вычисления. Для этого в программе необходимо наличие выражений. Выражения строятся из переменных, констант, функций и операций.

Переменная — это объект программы, который имеет имя и значение, причем имя переменной уникально и неизменно, а значение может меняться в процессе выполнения программы.

Переменные создаются программистом при написании программы. Присвоение имени переменной происходит по определенным правилам:

- первым символом имени должна быть буква;

- остальные символы — буквы или цифры (причем прописные и строчные символы различаются);
- число символов не должно превышать 255;
- имя не должно быть ключевым (зарезервированным) словом VB;
- нельзя использовать точку;
- имя переменной определяет область памяти (адрес), в которой будут храниться ее значения.

Значение переменной — это данные, которые хранятся и обрабатываются компьютером. Хранятся и обрабатываются данные по-разному. Это зависит от того, к какому типу они принадлежат.

Типом данных называется способ хранения и представления данных в компьютере. Тип переменных определяется диапазоном

Таблица 8.2

Типы данных VB

Тип данных	Возможные значения	Объем занимаемой памяти	Приставка к имени	Суффикс
Byte	Целые числа от 0 до 255	1 байт	byt	—
Integer	Целые числа от -32 768 до 32 767	2 байт	int	%
Long	Целые числа двойной длины от -2 147 483 648 до 2 147 483 647	4 байт	lng	&
Single	Десятичные числа одинарной точности от 1,401298E - 45 до 3,4022823E + 38	4 байт	sng	!
Double	Десятичные числа двойной точности от 4,94065645841247E - 324 до 4,79769313486232E308	8 байт	dbl	#
Boolean	Логическое значение True или False	1 байт	bln	—
String	Строка символов	1 байт на каждый символ	str	S
Cur	Число в денежном формате	8 байт	cur	@
Date	Дата от 1 января 100 г. до 31 декабря 9999 г.	8 байт	dtm	—
Object	Ссылки на любой объект	4 байт	obj	—
Variant	Любые значения	16 байт	vnt	—

данных, в котором могут изменяться значения переменных (табл. 8.2).

Значениями переменных числовых типов (Byte, Integer, Long, Single, Double) являются числа, логических (Boolean) — True или False, строковых (String) — последовательности символов и т.д. Обозначения типов переменных являются ключевыми словами языка и поэтому выделяются.

8.3.1. Объявление переменных и присваивание им значений

Важно, чтобы не только программист понимал, переменные какого типа используются в программе, но чтобы это мог учесть и исполнитель программы (компьютер). Последнее даже более важно, так как если компьютер не будет знать, переменная какого типа используется в программе, то он будет считать ее переменной универсального типа Variant и отведет для ее хранения в памяти 16 или более ячеек. Это будет приводить к неэффективному использованию памяти и замедлению работы программы, а иногда к ошибкам в работе. Поэтому, прежде чем работать с переменными, их необходимо объявить (указать) в программе. Объявить тип переменной можно разными способами.

Простейшим способом задания типа переменной (ее объявления) является приписывание к имени переменной определенного суффикса. Например, числовую переменную типа Integer можно задать как A%, а строковую переменную String — как A\$ (см. табл. 8.2).

Для объявления типа переменной можно воспользоваться также оператором определения переменной. Понятие оператора является одним из основных в языках программирования.

Программа, как мы уже знаем, представляет собой описание алгоритма решения задачи, которое должно быть воспринято исполнителем этого алгоритма. Алгоритм состоит из отдельных действий — предписаний.

Оператор — это такая синтаксическая единица языка программирования, которая используется в программе для выполнения отдельных предписаний. Любая программа состоит из последовательности операторов, которые записываются в соответствии с синтаксическими правилами. Итак, оператор определения переменной записывается с помощью строки программного кода.

Синтаксис этого оператора следующий:

Dim ИмяПеременной [**As** ТипПеременной]

Здесь «Dim» и «As» — ключевые слова языка Visual Basic. Назначение этого оператора — объявить переменную, т.е. задать ее имя и тип, однако объявление типа может отсутствовать. С помо-

щью одного оператора можно объявить сразу несколько переменных, например:

Dim Число As Integer, Строка As String, номер1, номер2 As byte
Переменные, значения которых не меняются в процессе выполнения программы, называются *константами*. Синтаксис объявления констант следующий:

Const ИмяКонстанты [As Тип] = ЗначениеКонстанты

Объявление переменных и констант происходит в начале программы.

Во всех языках программирования есть оператор, с помощью которого можно сделать пояснения в программе — это комментарии. Комментарии никак не влияют на ход выполнения программы (полностью игнорируются в процессе ее выполнения) и нужны только для того, чтобы было легче разбираться в программе.

В Visual Basic комментарий — это текст произвольного размера, который слева помечается символом «'» (апострофа). При записи программы в окне *Code* комментарии автоматически выделяются зеленым цветом.

Переменная может получить или изменить значение с помощью оператора присваивания. Синтаксис оператора следующий:

[Let] ИмяПеременной = Выражение

Например, после выполнения фрагмента программы

Dim A, B As Integer, C As Single

A = 7

B = 4

C = (A + B)/(A - B)

вещественная переменная C примет значение, равное числу 366 666,67.

После выполнения фрагмента программы

Dim A, C As boolean

A = "форма"

C = "ин" + A + "тика"

строковая переменная C примет значение, равное строке "ин-форматика".

После выполнения фрагмента программы

A = 5 > 3

B = 2*2 = 5

C = A And B

логическая переменная C примет значение False (ЛОЖЬ), так как A имеет значение True (ИСТИНА), а B — значение False (ЛОЖЬ).

Над различными типами данных, а следовательно, переменных, допустимы различные операции. Так, над числовыми переменными возможны арифметические операции, над логическими — логические операции, над строковыми — операции преобразова-

ния символьных строк и т.д. Различные типы данных требуют для своего хранения в оперативной памяти компьютера различное число ячеек (байт). Так, для хранения целых чисел в интервале от 0 до 255 в переменных типа Byte достаточно одной ячейки памяти (1 байт), для хранения вещественного числа с двойной точностью в переменных типа Double требуется уже восемь ячеек (8 байт), а для хранения символьных строк в переменных типа String требуется одна ячейка на каждый символ. Рассмотрим, как создать программу, которая выполняет арифметические операции над числами.

Пример 1. Создадим программу, которая выполняет деление двух целых чисел, а для хранения результата будем использовать различные типы числовых переменных, т.е. результат будет вычисляться с различной точностью.

Пусть в качестве аргументов программы выступают две целочисленные переменные A и B, а в качестве результатов — целочисленная переменная C, переменная одинарной точности D и переменная двойной точности E. Прежде всего необходимо объявить эти переменные для их использования в программе:

```
Dim A, B As Integer, C As Byte, D As Single, E As Double
```

Для создания графического интерфейса разместим на форме (Form1) управляющий элемент «Командная кнопка» (Command1). В качестве события, которое будет запускать программный модуль на выполнение, выберем щелчок по командной кнопке, где объект — Command1, а событие — Click(). Таким образом, мы создали заготовку событийного программного модуля:

```
Dim A, B As Integer, C As Byte,  
D As Single, E As Double  
Private Sub Command1_Click()  
End Sub
```

Теперь необходимо разработать программный модуль, который должен реализовать алгоритм деления двух чисел:

1) присвоить аргументам алгоритма (переменным A и B) конкретные значения;

2) присвоить переменным C, D и E результат деления аргументов;

3) напечатать результаты (значения переменных) на форме.

Реализацию первых двух инструкций (команд) алгоритма произведем с помощью операторов присваивания. В качестве исходных чисел возьмем целые числа 2 и 3:

```
A = 2  
B = 3  
C = A / B  
D = A / B  
E = A / B
```

Печать результатов выполнения программного модуля осуществим с помощью метода Print, которым обладает объект «Form1». Метод Print используется для печати на форме чисел и строк, а также значений числовых и строковых переменных или выражений, которые образуют список печати.

В качестве разделителей списка печати используется либо запятая, либо точка с запятой. В первом случае элементы списка печатаются каждый в своей зоне (по 14 символов в каждой зоне), во втором случае элементы списка печатаются вплотную друг к другу. В случае отсутствия списка печати на форму выводится пустая строка.

Синтаксис метода Print следующий:

Объект.Print [СписокПечати]

В нашем случае *СписокПечати* образуют переменные, значения которых необходимо напечатать:

Form1.Print C, D, E

Теперь мы можем записать программу целиком:

Dim A, B As Integer, C As Byte, D As Single, E As Double

Private Sub Command1_Click()

A = 2

B = 3

C = A / B

D = A / B

E = A / B

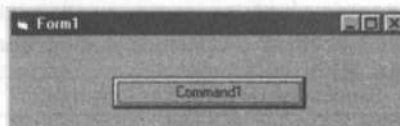
Form1.Print C, D, E

End Sub

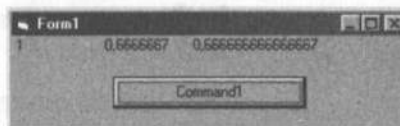
После запуска программы на экране появится окно с размещенной на ней командной кнопкой *Command1*. Щелчок по кнопке *Command1* (рис. 8.5, *а*) вызовет выполнение событийного программного модуля и на форме будут напечатаны результаты реализации алгоритма (рис. 8.5, *б*).

Результаты показывают, что деление двух чисел производится с различной точностью, которая зависит от типа переменной, используемой для хранения результатов вычислений (см. гл. 2).

После запуска программы на выполнение процессор последовательно будет считывать из памяти операторы и выполнять их. В процессе выполнения оператора объявления переменных



а



б

Рис. 8.5. Формы в VB:

а — исходная; *б* — после выполнения программы

Dim в оперативной памяти для их хранения отводится необходимое число ячеек. Для целочисленных переменных A, B — по две ячейки; для целочисленной переменной C — одну ячейку; для переменной одинарной точности D — четыре ячейки; для переменной двойной точности E — восемь ячеек. Таким образом, в памяти для хранения данных (значений переменных) будет отведено 17 ячеек (например, ячейки с 1-й по 17-ю). Далее, в процессе выполнения операторов присваивания, в отведенные переменным области оперативной памяти записываются их значения. Если в правой части оператора присваивания находятся арифметические выражения, то предварительно вычисляют их значения. Затем, с помощью метода Print производится вывод значений переменных на форму, реализующую графический интерфейс программы. В этом процессе значения переменных считываются из памяти и высвечиваются на экране монитора.

8.3.2. Выражения

Переменные могут образовывать арифметические, строковые и логические выражения.

Арифметические выражения. В состав арифметических выражений могут входить переменные числового типа и числа. Над переменными и числами могут производиться различные арифметические операции, а также математические операции, выражения с функциями (табл. 8.3).

Результатом целочисленного деления одного числа на другое является целая часть их отношения (например, значение выражения $23 \setminus 5$ равно 4). Результатом деления по модулю одного числа на другое является остаток целочисленного деления (например, значение выражения $23 \bmod 5$ равно 3).

Таблица 8.3

Некоторые арифметические операции

Запись арифметических выражений	Арифметические операции
$A + B$	Сложение A и B
$A - B$	Вычитание B из A
$A * B$	Умножение A на B
A / B	Деление A на B
$A \setminus B$	Целочисленное деление A на B
$A ^ B$	Возведение A в степень B
$-A$	Перемена знака A
$A \bmod B$	Деление по модулю A на B

Вычисление арифметических выражений производится в соответствии с общеизвестным порядком выполнения арифметических операций (возведение в степень, умножение или деление, сложение или вычитание), который может изменяться с помощью скобок.

Например, известная из курса физики формула для вычисления пути при равноускоренном движении тела в форме арифметического выражения примет следующий вид:

$$S = VT + \frac{AT^2}{2}.$$

Строковые выражения. В состав строковых выражений смогут входить переменные строкового типа, строки и строковые функции.

Строками являются любые последовательности символов, заключенные в кавычки, например объявление:

"информатика", "2000", "2*2".

Над переменными и строками может производиться операция конкатенации (объединения). Операция конкатенации заключается в том, что она объединяет строки или значения строковых переменных в единую строку. Операция конкатенации обозначается знаком **++**, который не следует путать со знаком сложения чисел в арифметических выражениях.

Логические выражения. В состав логических выражений кроме логических переменных могут входить также числа, числовые или строковые переменные или выражения, которые сравниваются между собой с использованием операций сравнения (**>**, **<**, **=**, **>=**, **<=** и т.д.) (см. подразд. 7.2).

Над элементами логических выражений могут производиться логические операции, которые на языке Visual Basic обозначаются следующим образом: логическое умножение — **And**, логическое сложение — **Or**, логическое отрицание — **Not**. При записи сложных логических выражений часто используются скобки, например:

(5 > 3) And (2*2 = 5) — условие ложно;

(5 > 3) Or (2*2 = 5) — условие истинно.

8.3.3. Функции в языке Visual Basic

В определении выражения присутствует объект «Функция».

Понятие функции в языках программирования близко к понятию функции в математике. Функция может иметь один или более аргументов, если функция имеет один аргумент, то она является

функцией одного аргумента. Синтаксис записи функции нескольких аргументов следующий (аргументы в списке отделяются друг от друга запятыми):

ИмяФункции(СписокАргументов:аргум.1, аргум.2, ...)

Функции обычно входят в состав выражений, значение которых присваивается переменным. В VB функции бывают встроенные и системные. Встроенные функции могут быть различных типов: математические, строковые, финансовые, даты и др. Тип функции определяется возможными значениями аргументов и функций.

Функции ввода-вывода. Для ввода данных в программу через текстовое поле на диалоговой панели используется функция *InputBox* (окно ввода). В качестве аргументов этой функции выступают три строки, и значением функции является также строка. Синтаксис функции следующий:

InputBox(Приглашение\$, Заголовок\$, [ПоУмолчанию\$])

В процессе выполнения этой функции появляется диалоговая панель с текстовым полем. В строке заголовка панели печатается значение второго аргумента «Заголовок\$», на самой панели печатается значение аргумента «Приглашение\$», в текстовом поле печатается значение аргумента «ПоУмолчанию\$» (если это значение отсутствует, то содержимое текстового окна также отсутствует). Введенная пользователем в текстовом поле строка становится значением функции. Составим простейшую программу на использование функции *InputBox* (рис. 8.6, а).

```
Dim A As String
Sub Command1_Click()
    A = InputBox("Введите фамилию, имя, отчество:", "ОкноВвода")
    Form1.Print A
End Sub
```

Для вывода сообщений не на форме, а в специальном окне сообщений может использоваться функция *MsgBox* (окно сообщений). Синтаксис функции в режиме оператора использует запись аргументов без скобок:

MsgBox Сообщение\$, ЧисКод1+ЧисКод2[, Заголовок\$]

Строка «Сообщение\$» выводится на панели сообщений, аргумент «ЧисКод1+ЧисКод2» определяет внешний вид панели, а строка «Заголовок\$» печатается в строке заголовка панели. Последние два аргумента не обязательны. Текст может содержать до 1024 символов. (Для того чтобы этот текст выдавался не сплошной длинной строкой, его можно разбить на небольшие части, между которыми вставляются специальные «невидимые» символы — пере-

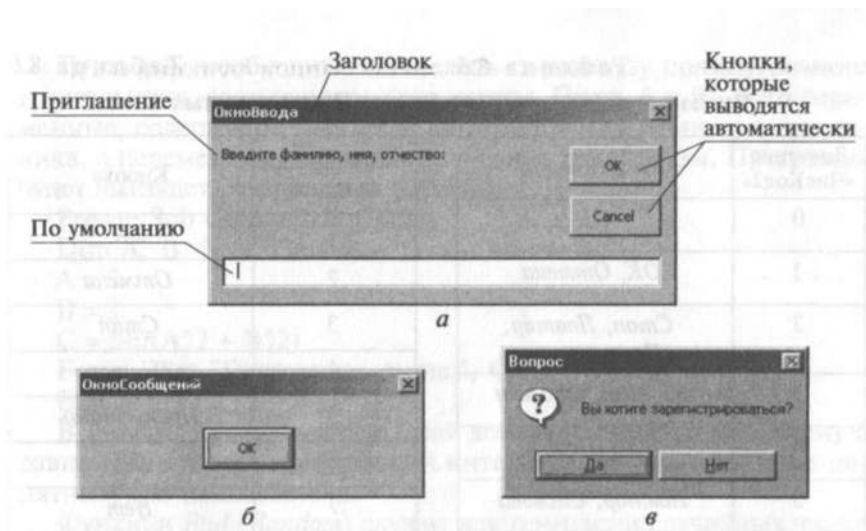


Рис. 8.6. Системные функции:

а — окно ввода InputBox; *б, в* — примеры окон сообщений MsgBox

нос и возврат к левому краю страницы.) Значение «ЧисКод1» определяет вид сообщения и пиктограмму, которая помещается в окно *MsgBox* (табл. 8.4).

Таких пиктограмм четыре. Они соответствуют четырем видам сообщений. Значение «ЧисКод2» определяет набор кнопок в окне *MsgBox* (табл. 8.5).

Вы легко можете убедиться в том, что для любой комбинации «ЧисКод1» и «ЧисКод2» их сумма будет уникальной, т.е. с помо-

Таблица 8.4

Соответствие системных сообщений числовым кодам

Значение «ЧисКод1»	Вид сообщения	Пиктограмма
16	Критическое сообщение	
32	Вопрос	
48	Предупреждение	
64	Информация	

Таблица 8.5 Таблица 8.6

Числовые коды

Значение «ЧисКод2»	Набор кнопок
0	<i>ОК</i>
1	<i>ОК, Отмена</i>
2	<i>Стоп, Повтор, Пропустить</i>
3	<i>Да, Нет, Отмена</i>
4	<i>Да, Нет</i>
5	<i>Повтор, Отмена</i>

Коды нажатых кнопок

Возвращаемое значение	Кнопка
1	<i>ОК</i>
2	<i>Отмена</i>
3	<i>Стоп</i>
4	<i>Повтор</i>
5	<i>Пропустить</i>
6	<i>Да</i>
7	<i>Нет</i>

шью одного числа вы можете установить и определенную пиктограмму, и определенную комбинацию кнопок.

Действие функции *MsgBox* в операторной форме заключается в том, что на экране появляется окно сообщений. После нажатия кнопки *ОК* в окне сообщений выполнение программы продолжается (см. рис. 8.6, б).

Если используется бесскбочная форма синтаксиса, то нажатие одной из кнопок на этом окне просто завершает работу функции, а если используется форма со скобками, то значение функции присваивается какой-нибудь переменной. Возвращаемое значение — это целое число от 1 до 7 (табл. 8.6). Оно зависит от того, какая из кнопок окна *MsgBox* нажата.

В качестве примера использования функции рассмотрим простейшую программу, выводящую на панель сообщений запрос на регистрацию пользователя (рис. 8.6, в).

Следует обратить внимание на необходимость вывода запятой вместо отсутствующего аргумента.

```
Private Sub Command1_Click()  
MsgBox "Вы хотите зарегистрироваться?", 32+4, "Вопрос"  
End Sub
```

В случае записи аргументов в скобках функция *MsgBox* получает определенное значение, которое может быть присвоено какой-либо переменной. Синтаксис функции в этом случае следующий:

```
MsgBox(Сообщение$, ЧисКод1+ЧисКод2)[, Заголовок$])
```

Математические функции. В математических функциях значениями как аргументов, так и функций являются числа. Всего в языке Visual имеется 12 математических функций, представленных в табл. 8.7.

Пусть нам необходимо вычислить гипотенузу прямоугольного треугольника, если известны его катеты. Пусть A и B — это переменные, содержащие значения катетов прямоугольного треугольника, а переменная C содержит значение гипотенузы. Программа будет выглядеть следующим образом:

```
Private Sub Command1_Click()
    Dim A, B, C As Double
    A = 2
    B = 3
    C = Sqr(A^2 + B^2)
    Form1.Print "Гипотенуза равна:", C
End Sub
```

В список печати метода Print добавлена строка «Гипотенуза равна:», что делает графический интерфейс программы более понятным для пользователя.

Функция *Rnd* (*Random*) служит для генерации случайных чисел: *Rnd* [(значение)]

Функция *Rnd* возвращает значение в диапазоне от 0 до 1 типа Single, содержащее случайное число (причем 1 не входит в этот диапазон, а 0 входит). Функция возвращает псевдослучайные числа. При каждом запуске программы функция генерирует одну и ту же последовательность случайных чисел. Во избежание этого явления используйте инструкцию *Randomize*.

Таблица 8.7

Математические функции

Функция	Возвращаемое значение
Sin(X)	Синус числа X
Cos (X)	Косинус числа X
Tan (X)	Тангенс числа X
Atn(X)	Арктангенс числа X
Sqr (X)	Квадратный корень из числа X
Log (X)	Логарифм числа X
Exp(X)	Показательная функция числа X
Int(X)	Наибольшее целое число не превышающее число X
CInt (X)	Целое число, ближайшее к числу X
Fix(X)	Целое число, равное числу X без дробной части
Abs(X)	Абсолютное значение числа X
Rnd(A)	Случайное число

Примечание: чтобы получить значения случайных чисел в интервале от min до max используйте формулу

$$\text{Int}((\text{max} - \text{min} + 1) * \text{Rnd} + \text{min}),$$

где min и max — соответственно максимальное и минимальное числа.

Необязательный аргумент представляет число типа *Single* или любое допустимое числовое выражение. Аргумент должен принимать следующие значения:

- аргумент опущен или число больше 0. Генерируется следующее случайное число в последовательности. При повторном запуске программы генерируется аналогичная последовательность случайных чисел;
- аргумент отрицательный. Генерируется одно и то же число, используя аргумент как опорное число;
- аргумент равен 0. Генерируется одно и то же число, используя число при предыдущем вызове функции.

Пример 2.

' Генерируем случайные числа от 1 до 7

```
Private Sub Command1_Click()
```

```
Dim A As Integer
```

```
A = Int((7*Rnd) + 1)
```

```
Print A ' выводим число на форме
```

```
End Sub
```

Строковые функции. Строковые функции позволяют выполнять обработку данных строкового типа.

Функция определения длины строки *Len*. В функции определения длины строки *Len*(Строка\$) аргументом является строка «Строка\$», а возвращает функция числовое значение длины строки (число символов в строке).

Синтаксис функции:

Len(Строка\$)

Пусть аргументом функции *Len* будет строка «информатика», тогда значением целочисленной переменной

ДлинаСтроки = *Len*("информатика")

будет число 11.

Функции вырезания подстроки *Left*, *Right*, *Mid*. В функциях вырезания подстроки (части строки) *Left* (Строка\$, Длина %), *Right* (Строка\$, Длина %) и *Mid* (Строка\$, Позиция %, Длина %) аргументами являются строка «Строка\$» и числа или целочисленные переменные «Длина %» и «Позиция %». Функции возвращают строковое значение, равное длине вырезанной подстроки.

Синтаксис функций:

Left(Строка\$, Длина %)

Right(Строка\$, Длина %)

Mid(Строка\$, Позиция %, Длина %)

Значением функции *Left* является левая подстрока, которая начинается от крайнего левого символа строки и имеет число символов, равное значению числового аргумента «Длина %».

Пусть аргументом функции *Left* будет строка «информатика», тогда значением строковой переменной

ЛеваяПодстрока = Left("информатика",2)

будет строка «ин». Значением функции *Right* является правая подстрока, которая начинается от крайнего правого символа строки и имеет число символов, равное значению числового аргумента «Длина %».

Пусть аргументом функции *Right* будет строка «информатика», тогда значением строковой переменной

ПраваяПодстрока = Right("информатика",4)

будет строка «тика».

Значением функции *Mid* является подстрока, которая начинается от позиции символа, заданной числовым аргументом «Позиция %», и длиной, равной значению числового аргумента «Длина %».

Пусть аргументом функции *Mid* будет строка «информатика», тогда значением строковой переменной

strПодстрока = Mid("информатика",3,5)

будет строка «форма».

Для преобразования текста из начального состояния «информационные технологии» в конечное состояние «технологии информационных» необходимо выполнить следующую программу:

```
Dim A, B, C As String
Private Sub Command1_Click()
A = "информационные технологии"
B = Left(strA, 14)
C = Right(strA, 10)
Form1.Print (B + C)
End Sub
```

Рекомендуем проверить правильность работы этой программы самостоятельно.

Функции преобразования типов данных. Функции преобразования реализуют преобразование данных из одного типа в другой.

Часто необходимо преобразовать строковое значение в числовое значение. Это можно сделать с помощью функции *Val*, аргументом которой является строка, а значением — число:

Val(Строка\$)

Например, значением функции *Val* (2000) является число 2000. Часто эта функция применяется для преобразования строкового значения функции *InputBox* в число, которое затем используется в арифметических выражениях.

Выше была рассмотрена программа вычисления гипотенузы прямоугольного треугольника. Усовершенствуем эту программу так, чтобы произвольные значения катетов пользователь мог вводить с клавиатуры. Реализуем это с помощью функции *InputBox*. Далее преобразуем строковые значения катетов в числовые с помощью функции *Val* и произведем вычисления:

```
Dim A, B As String, A, B, C As Double
Private Sub Command1_Click()
A = InputBox("Первый катет:", "Катет A")
B = InputBox("Второй катет:", "Катет B")
A = Val(strA)
B = Val(strB)
C = Sqr(A^2 + B^2)
Form1.Print "Гипотенуза равна:", dblC
End Sub
```

Строковое выражение числа может быть задано не только в десятичной, но и в восьмеричной, и в шестнадцатеричной системах счисления.

Функция *Str* осуществляет обратное преобразование, т.е. преобразование числа в строку. Аргументом функции является число, а значением — строка:

Str(Число)

Например, значением функции *Str*(2000) является строка «2000». Данная функция часто применяется для преобразования чисел в строки для их вывода на панель сообщений с помощью функции *MsgBox*.

В рассмотренной выше программе вычисления гипотенузы можно ее значение не печатать на форме, а выводить с помощью панели сообщений. Для этого строку программы, содержащую метод *Print* необходимо заменить на строку

MsgBox Val (C), , "Гипотенуза C равна"

Функции *Hex* и *Oct* позволяют производить преобразование десятичных чисел в шестнадцатеричные и восьмеричные числа в строковой форме.

```
Private Sub Command1_Click()
```

```
form1.Print Hex (2000), Oct (2000)
```

```
End Sub
```

Функция *Asc* осуществляет преобразование строки в числовой код первого символа. Аргументом функции является строка, а значением — число:

```
Asc(Строка$).
```

Например, значением функции *Asc*("Э") является числовой код символа 221. Программа, реализующая преобразование символа и печать его кода, приведена ниже:

```
Dim A As Byte
Private Sub Command1_Click()
A = Asc("Э")
Form1.Print A
End Sub
```

Функция *Chr* осуществляет преобразование числового кода в символ. Аргументом функции является число, а значением — символ:

```
Chr(Число)
```

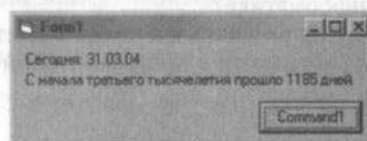
Например, значением функции *Chr*(221) является символ «Э». Программа, реализующая преобразование кода символа в символ, приведена ниже:

```
Dim A As String
Private Sub Command1_Click()
A = Chr(221)
Form1.Print A
End Sub
```

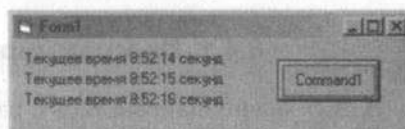
Функции даты и времени. Функция *Date* возвращает значение текущей даты, которое можно присвоить переменным типа *Date*. Значение даты представляется в виде чисел «Месяц/Число/Год», разделенных знаком «/». Разностью значений переменных типа *Date* является число дней между датами.

Можно, например, составить программу, печатающую текущую дату и число дней, прошедших с начала третьего тысячелетия (рис. 8.7, а):

```
Dim A, B As Date
Private Sub Command1_Click()
A = Date
B = #1/1/2001#
```



а



б

Рис. 8.7. Функции даты и времени:
а — программа вывода текущей даты;
б — программа вывода текущего времени

```

Form1.Print "Сегодня"; A
Form1.Print "С начала третьего тысячелетия прошло"; A — B;
"дней"
End Sub

```

Функция *Time* возвращает значение текущего времени, которое можно присвоить переменным типа *Long*. Значение времени представляется в виде чисел «Часы:Минуты:Секунды», разделенных знаком «:». Можно, например, составить программу, определяющую текущее время (рис. 8.7, б). Значение текущего времени печатается после нажатия на кнопку *Command1*.

```

Dim A As Time
Private Sub Command1_Click()
A = Time
Form1.Print "текущее время"; A; "секунд"
End Sub

```

В языке Visual Basic существуют и другие встроенные функции, однако рассмотреть их все в данном учебном пособии не представляется возможным.

Контрольные вопросы

1. Какие типы данных используются в VB?
2. В чем разница между типом, именем и значением переменной?
3. Какие объекты могут входить в состав арифметических, строковых и логических выражений?
4. Почему рекомендуется объявить переменные перед их использованием в программе?
5. В чем заключается суть операции присваивания?
6. Чем отличается понятие функции в математике и понятие функции в информатике?
7. Какой диапазон чисел может храниться в переменной типа *Long*?
8. Можно ли в программе обойтись без констант?

8.4. ПРОГРАММИРОВАНИЕ ВЕТВЛЕНИЙ И ПОВТОРЕНИЙ

8.4.1. Программирование ветвлений

В гл. 7 мы познакомились с разветвляющимися алгоритмами. Для программирования логических выражений используют структуры ветвления. Напомним, что в алгоритмической структуре ветвления те или иные операторы выполняются в зависимости от истинности условия. Структуру ветвления называют оператором условного перехода, который может быть организован одним из следующих способов:

- многострочная форма;
- однострочная форма;
- функция *If*.

В многострочной форме этот оператор записывается с помощью следующей инструкции:

```
If <условие> Then
<операторы1>
Else
<операторы2>
End If
```

В этом случае второе ключевое слово «Then» размещается на той же строчке, что и условие, а последовательность команд (операторы1) должна размещаться на следующей строчке. Третье ключевое слово «Else» размещается на третьей строчке, а последовательность команд (операторы2) — на четвертой. Конец инструкции ветвления «End If» размещается на пятой строчке.

В однострочной форме оператор условного перехода записывается с помощью инструкции

```
If <условие> Then <операторы1> Else <операторы2>
```

Если инструкция не помещается на одной строке, то она может быть разбита на несколько строк. Такое представление инструкции более наглядно для человека. Компьютер же должен знать, что разбитая на строки инструкция представляет единое целое. Это обеспечивает знак переноса, который задается символом «Подчеркивание» после пробела.

Третье ключевое слово «Else» в сокращенной форме инструкции может отсутствовать (необязательные части операторов записываются в квадратных скобках). Если условие ложно, то выполнение оператора условного перехода заканчивается и выполняется следующая строка программы.

Функция *If* записывается с помощью инструкции

```
If (<условие>, <значение1>, <значение2>)
```

Функция *If* возвращает значение1, если условие истинно, и значение2, если условие ложно.

Пример 1. Ввести значения *A* и *B*, если $A > B$, присвоить переменной *C* значение суммы квадратов чисел *A* и *B*, в противном случае — значение *B*. Алгоритм решения представлен на рис. 8.8, а.

Разместим на форме следующие объекты:

- Label1 — со свойством Caption, имеющим значение «*A*»;
- Label2 — со свойством Caption, имеющим значение «*B*»;
- Text1 — для ввода значения *A* через свойство *Text*;
- Text2 — для ввода значения *B* через свойство *Text*;

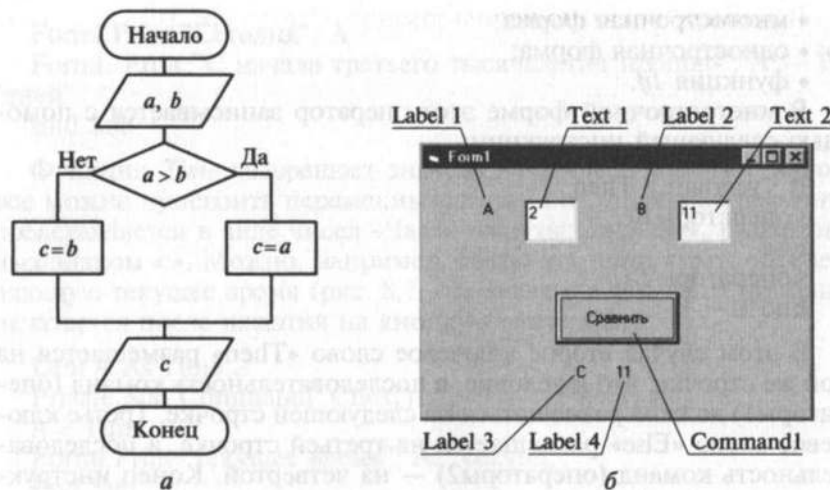


Рис. 8.8. Программирование ветвлений:
а — алгоритм; б — выполнение программы

Command1 — командная кнопка;

Label 3 — со свойством *Caption*, имеющим значение «С»;

Label 4 — для вывода значения переменной через свойство *Caption*.

Событийная процедура, которая выполняется при нажатии на кнопку *Сравнить*, выглядит следующим образом (результат работы приложения представлен на рис. 8.8, б):

```
Dim a, b, c As Integer
Private Sub Command1_Click()
    a = Val(Text1.Text)
    b = Val(Text2.Text)
    If a > b Then c = a^2+b^2 Else c = b
    Label4.Caption = c
End Sub
```

Обратите внимание на то, что переменные A, B и C типа Integer.

При использовании функции *Iif* событийная процедура будет выглядеть следующим образом:

```
Dim a, b, c as Integer
Private Sub Command1_Click()
    a = Val(Text1.text)
    b = Val(Text2.text)
    c = Iif (a>b, a^2+b^2, b)
    Label4. Caption = c
End Sub
```

В примере показано использование объекта «Метка» для ввода результата через свойство *Caption*.

Пример 2 (ввод пароля). Исходная форма содержит три текстовых поля: фамилию, имя и пароль. Оригинал пароля (строка символов) задается внутри событийной процедуры через переменную *paset*. Пароль, введенный в текстовом поле *Text3* (рис. 8.9, *а*), сравнивается с оригиналом. Если он верен, то выводится окно сообщения, представленное на рис. 8.9, *б*, в противном случае — окно сообщения, представленное на рис. 8.9, *в*.

Сама событийная процедура имеет следующий вид:

```
Dim fam, im, pas, paset As String
Private Sub Command1_Click()
    fam = Text1.Text
    im = Text2.Text
    pas = Text3.Text
    paset = "пароль"
    If pas = paset
    Then
        MsgBox ("Здравствуйте, " + fam + " " + im)
    Else
        MsgBox ("Пароль неверен!")
    End If
End Sub
```

В языке VBA есть еще один оператор, с помощью которого можно изменить последовательный ход выполнения программы. Это оператор безусловного перехода *Go To* («перейти к ...»). Обычно его называют безусловным оператором и его синтаксис выглядит следующим образом:

Go To Метка

Метка — символьная цепочка, которая может быть поставлена перед каким-либо оператором программы (с этого оператора про-

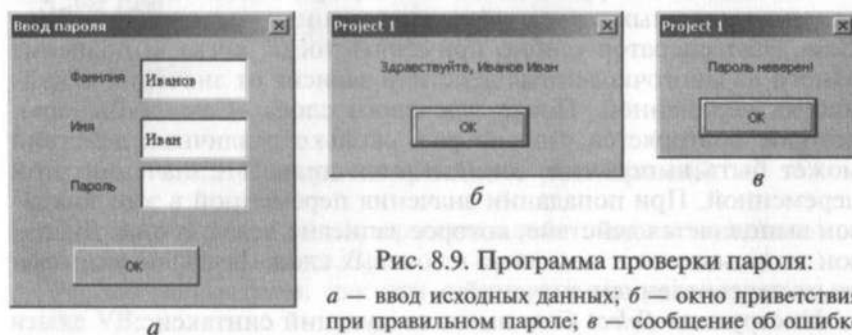


Рис. 8.9. Программа проверки пароля:

а — ввод исходных данных; *б* — окно приветствия при правильном пароле; *в* — сообщение об ошибке

грамма продолжает свою работу). Переход к этому оператору называется безусловным переходом. В отличие от имени переменной метка может начинаться с цифры.

Пример 3. Даны три переменные a , b , c , которые являются длинами отрезков. Можно ли из этих отрезков построить треугольник? Задачу решим двумя способами: с использованием условного оператора и оператора безусловного перехода.

Для того чтобы из отрезков a , b и c можно было построить треугольник, необходимо выполнение (одновременное) трех неравенств:

$$a + b > c; b + c > a; a + c > b.$$

Условный оператор	Условный оператор + Go To
...	...
If (a+b) >= c And	If (a+b) <= c Go To M1
(b+c) >= a And	Else
(a+c) >= b	If (a+c) <= b Then Go To M1
Then	Else
MsgBox("Треугольник построить можно")	If (b+c) <= Then Go To M1
Else	MsgBox("Треугольник построить можно")
MsgBox("Треугольник построить нельзя")	Go To M2
...	M1: MsgBox("Треугольник построить нельзя")
	M2:
	...

Хорошим стилем программирования считается такой, в котором при написании программы не используется оператор *Go To*. Действительно, в основном многие задачи программируются без использования безусловного оператора.

Еще одной инструкцией языка VB, предназначенной для реализации сложных ветвлений, является инструкция выбора *Select Case*. Этот оператор удобно применять тогда, когда выполнение одного из многочисленных действий зависит от значения какой-нибудь переменной. После ключевого слова «Case» («Выбор»), которое повторяется столько раз, сколько различных действий может быть выполнено, записывается диапазон значений этой переменной. При попадании значения переменной в этот диапазон выполняется действие, которое записано вслед за ним. Диапазон записывается с помощью ключевых слов «Is», «To» или простым перечислением значений.

Инструкция *Select Case* имеет следующий синтаксис:

```

Select Case <переменная>
Case <значение1>
<операторы1>
...
Case <значение(N-1)>
<операторы(N-1)>
[Case Else
[<операторыN>]
End Select

```

В качестве значения для блока Case можно указывать не только одно значение (1), но и несколько значений, разделенных запятой (2, 3). Можно определять также области сравнения (4 *To* 6) или воспользоваться относительным сравнением (Is >= 9). Вместо непосредственного значения проверочного выражения можно использовать ключевое слово «Is».

Блок Case Else выполняется, если ни одно из предыдущих условий не является истинным. Если условию Select Case соответствует несколько блоков Case, то выполняется первый из них.

Пример 4. По номеру месяца определить, к какому времени года он относится. Использование оператора Select Case позволяет компактно записать программу этого приложения:

```

Dim A As Integer
Private Sub Command1_Click()
A = Val(InputBox("Введите месяц :", "Время года"))
Select Case A
Case 12,1,2
Form1.Print "Зима"
Case 3 To 5
Form1.Print "Весна"
Case 6 To 8
Form1.Print "Лето"
Case 9 To 11
Form1.Print "Осень"
Case Else
Form1.Print "Введено неверное значение"
End Select
End Sub

```

8.4.2. Программирование повторений

Для многократно повторяющегося действия в алгоритмах используются циклы (см. гл. 7).

Теперь рассмотрим, как эти структуры программируются на языке VB.

Циклические алгоритмические структуры бывают двух типов:

- циклы со счетчиком, в которых тело цикла выполняется определенное число раз;
- циклы с условием, в которых тело цикла выполняется до тех пор, пока выполняется условие.

Цикл с параметром. Когда заранее известно, какое число повторений тела цикла необходимо выполнить, можно воспользоваться циклической инструкцией (оператором цикла со счетчиком) *For ... Next*.

```
For Счетчик = НачЗнач To КонЗнач [<step шаг>]
<тело цикла>
Next <счетчик>
```

Синтаксис оператора *For ... Next* следующий: строка, начинающаяся с ключевого слова «For», является заголовком цикла, а строка с ключевым словом «Next» — концом цикла. Между ними располагаются операторы, являющиеся телом цикла. В начале выполнения цикла значение переменной *Счетчик* устанавливается равным *НачЗнач*. При каждом «проходе» цикла переменная *Счетчик* увеличивается на величину шага. Если она достигнет величины *КонЗнач*, то цикл завершается и выполняются следующие за ним операторы.

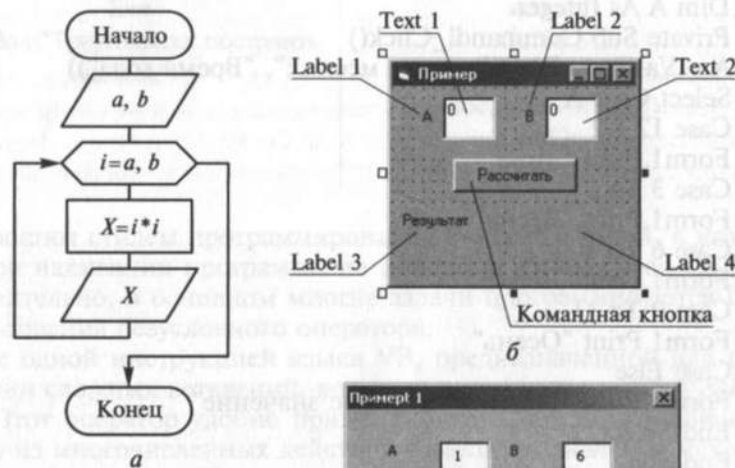


Рис. 8.10. Программирование циклической структуры:

a — алгоритм; *б* — форма; *в* — результат

Пример 1. Необходимо вычислить квадраты чисел в диапазоне от a до b . Значения a и b определяет пользователь. Алгоритм представлен на рис. 8.10, а). Для решения задачи поместим на форму два поля для ввода a и b , три поясняющие кнопки, одно поле вывода и кнопку *Рассчитать* (рис. 8.10, б).

Процедура обработки события нажатия на кнопку *Рассчитать* выглядит следующим образом:

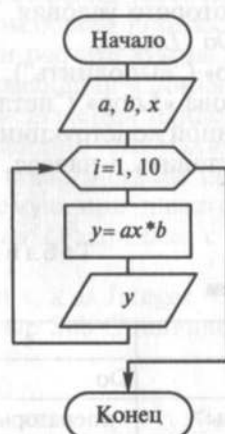
```
Private Sub Command1_Click()
    Dim i, a, b As Integer
    Dim s As String
    a = Val(Text1.Text)
    b = Val(Text2.Text)
    s = " "
    For i = a To b
        Label4.Caption = Label4.Caption + s + Str(i * i)
    Next
End Sub
```

Сделаем некоторые пояснения:

Оператор $s = " "$ означает, что строковый переменной s присваивается пустая строка (пробел).

« $\text{Label4.Caption} = \text{Label4.Caption} + s + \text{Str}(i * i)$ » означает, что в свойстве *Caption* происходит накопление значений квадратов чисел, которые отделяются друг от друга пробелом. Результат работы программы представлен на рис. 8.10, в.

Пример 2. Необходимо рассчитать 10 значений функции $y = ax + b$. Значения коэффициентов a и b , а также начальное значение x определяет пользователь. Алгоритм представлен на рис. 8.11, а.



б

Рис. 8.11. Программирование цикла с параметром:

а — алгоритм; б — форма

Разместим на форме следующие элементы:

Text1 — для ввода значений коэффициента a ;
 Text2 — для ввода значений коэффициента b ;
 Text3 — для ввода начального значения x ;
 Label1, Label2, Label3, Label4, Label5 — для заголовков;
 Label6 и Label7 — для вывода значений x и функции y .

Расчет значений будет происходить при нажатии на кнопку *Рассчитать*. Обработка событий при нажатии на кнопку выглядит следующим образом:

```
Private Sub Command1_Click()
  Dim i, a, b, x As Integer
  Dim s As String
  a = Val(Text1.Text)
  b = Val(Text2.Text)
  x = Val(Text3.Text)
  s = " "
  For i = 0 To 9
    Label6.Caption = Label6.Caption + s + Str(x + i)
    Label7.Caption = Label7.Caption + s + Str(a * (x + i) + b)
  Next i
End Sub
```

Для данного примера исходной точкой является $x = 2$, поэтому цикл, который обеспечивает приращение значения x , будет изменяться в интервале от 0 до 9. Через свойство *Caption* метки Label6 выводится текущее значение x , а через свойство *Caption* метки Label7 — значения y .

Цикл с условием. Часто бывает так, что необходимо повторить тело цикла, но заранее неизвестно, сколько раз это надо сделать, так как число повторений зависит от некоторого условия. Такой цикл реализуется с помощью оператора *Do Loop*.

Цикл начинается с ключевого слова «Do» ("выполнить"), а последняя строка начинается с ключевого слова «Loop» ("петля").

Существуют четыре разновидности заданной конструкции (табл. 8.8). Условие выхода из цикла можно поставить в начале, перед

Таблица 8.8

Виды циклов с условием

1	2	3	4
Do While ...	Do Until ...	Do	Do
<операторы>	<операторы>	<операторы>	<операторы>
Loop	Loop	Loop While ...	Loop Until ...

телом цикла (такой цикл называется циклом с предусловием), или в конце (такой цикл называется циклом с послеусловием).

Проверка условия выхода из цикла проводится с помощью ключевых слов «While» или «Until». Эти слова придают одному и тому же условию противоположный смысл. Ключевое слово «While» («до тех пор») обеспечивает выполнение цикла до тех пор, пока выполняется условие, т.е. пока условие имеет значение «истина». Как только условие примет значение «ложь», выполнение цикла закончится. В этом случае условие является условием продолжения цикла.

Ключевое слово «Until» («пока») обеспечивает выполнение цикла до тех пор, пока не выполняется условие, т.е. пока условие имеет значение «ложь». Как только условие примет значение «истина», выполнение цикла закончится. В этом случае условие является условием завершения цикла.

В случае 1 и 3 используется условие типа «While» — продолжение цикла. Повторяющиеся операторы выполняются, если значение условного выражения равно True (ИСТИНА). В противном случае цикл завершается.

В случае 2 и 4 используется условие типа «Until» — завершение цикла. Повторяющиеся операторы выполняются, если значение условного выражения равно False (ЛОЖЬ). В противном случае цикл завершается.

Если условие стоит в первой строке, то может случиться так, что повторяющиеся операторы так ни разу и не будут выполнены. Иногда это бывает полезно.

Пример 3. Пусть необходимо подсчитывать количество чисел, вводимых пользователем с клавиатуры до тех пор, пока их сумма не станет равна 1000. На форме разместим следующие объекты:

- командная кнопка *Command1* для ввода последовательности чисел и расчета суммы вводимых чисел и их количества;
- командная кнопка *Command2* для вывода в текстовое поле «Text1» значения найденной переменной *k*.

Очевидно, что наличие двух командных кнопок реализуется двумя событийными процедурами. Событийную процедуру, выполняемую при нажатии пользователем на командную кнопку *Command1*, запишем с использованием цикла с послеусловием *While*.

```
Dim x, k as Integer
Private Sub Command1_Click()
    S = 0
    k = 0
    Do
        x = Val(InputBox("Число x:", ""))
        k = k + 1
```

Таблица 8.9

Ошибки компиляции в управляющих конструкциях

Системное сообщение	Описание
Block If without End If	Отсутствие End If для блока If
Else without If	Использование Else без If
End If without block If	Отсутствие If для блока End If
For without Next	Использование For без Next
Next without For	Использование Next без For
Do without Loop	Использование Do без Loop
Loop without Do	Использование Loop без Do

```

s = s + x
Loop While S <= 1000
End Sub
Private Sub Command2_Click()
Text1.Text = Str(k)
End Sub

```

Тот же самый пример рассмотрим с использованием цикла с послеусловием *Until*.

```

Private Sub Command1_Click()
S = 0
k = 0
Do
x = Val(InputBox("Число x:", " "))
s = s + x
k = k + 1
Loop Until S >= 1000
End Sub

```

Событийная процедура, выполняемая при нажатии на кнопку *Command2*, аналогична приведенной в первом случае.

Несоответствие числа служебных слов для начала и окончания различных разновидностей структур ветвления и циклов приводит к возникновению ошибки на этапе компиляции с выводением стандартного сообщения.

В табл. 8.9 представлены возможные системные сообщения при возникновении ошибок компиляции.

Контрольные вопросы

1. Как происходит выполнение алгоритмической структуры «ветвление»?
2. Какие существуют формы записи оператора условного перехода?
3. Какие типы циклов существуют?

4. Можно ли реализовать пример 3 с использованием цикла с предусловием?
5. Приведите пример программы с безусловным оператором *Go To*, работа которой не может закончиться.
6. Что произойдет, если в цикле с предусловием знак неравенства заменить равенством?
7. Как с помощью функции *И* вычислять модуль числа?
8. Разработайте приложение, в котором по номеру месяца (1—12) будет выводиться его название (январь—декабрь).
9. Разработайте приложение, осуществляющее ввод последовательности из *n* чисел и подсчет количества чисел, кратных 7.
10. Разработайте приложение для вычисления значения $\sin x + \sin^2 x + \dots + \sin^n x$, не используя операции возведения в степень.

8.5. МАССИВЫ

Существует класс задач, для решения которых необходимо использовать множество переменных одного типа (например, для хранения фамилий и оценок студентов одной группы или всего колледжа; ежедневных значений температуры воздуха, если ведутся ее измерения в течение длительного времени).

Обработка таких данных может заключаться в расчете среднего балла всех студентов или среднего значения температуры воздуха в каждом месяце. Эти переменные необходимы для кратковременного хранения данных в оперативной памяти компьютера в течение работы программы с целью их дальнейшей обработки и называются переменной с индексом или массивом (см. подразд. 7.2).

Рассмотрим примеры использования массивов в программировании некоторых задач в среде проектирования Visual Basic.

Начнем с определения самого простого из массивов — одномерного массива (кроме одномерных существуют многомерные массивы, в частности, двумерные, или матрицы).

Одномерный массив — это конечное множество элементов одного типа.

В качестве элементов массива можно использовать любой ранее описанный тип, который называют в этом случае базовым типом.

Элементы, образующие массив, упорядочены таким образом, что каждому элементу соответствует индекс, определяющий его положение в общей последовательности. Доступ к каждому отдельному элементу осуществляется путем индексирования элементов массива.

Индекс — любое выражение, значением которого является целое число.

Объявление массива аналогично объявлению переменной, только кроме имени переменной надо указать в скобках диапазон значения индекса — минимальное и максимальное его значения,

между которыми записывается ключевое слово «То». Для двухмерного массива указываются два диапазона через запятую. Например:

```
Dim A (1 To 30) as Single  
Dim B (1 To 10) as Integer  
Dim D (1 To 5, 2 To 10) as Byte
```

Часто минимальное значение индекса не указывается. Тогда оно автоматически (по умолчанию) принимается равным нулю. Ключевое слово «То» при этом не используется. Но максимальное значение индекса должно быть указано обязательно. Например:

```
Dim C (30) as Byte
```

Для обращения к элементу массива необходимо указать имя массива и в скобках указать значение индекса. Например:

```
A(15), B(9), D(4,5).
```

Рассмотрим на примере, как происходит обработка массивов в программе.

Пример 1. Пусть массив содержит значения среднесуточной температуры воздуха в мае месяце 2004 г. Необходимо определить минимальную, максимальную и среднюю температуры воздуха в этом месяце.

Алгоритм нахождения максимальных и минимальных значений основан на выдвижении двух гипотез:

- 1) максимальным элементом массива является его первый элемент;
- 2) минимальным элементом массива является его первый элемент.

Эти гипотезы выражаются так: одно и то же значение «Температура(1)» присваивается двум переменным Max и Min.

Поиск максимального элемента массива рассмотрен в примере подразд. 7.2.

Алгоритм решения данной задачи приведен на рис. 8.12, а.

Для определения среднего значения температуры за месяц в программе необходимо вычислить сумму температур и разделить ее на число значений.

Программа на этапе выполнения представлена на рис. 8.12, б.

Событийная процедура, вызываемая при щелчке по кнопке *Расчет*, выглядит следующим образом:

```
Dim Max As Single, Min As Single, Aver As Single, Sum As Single,  
i As Single, Temp(1 To 30) As Single  
Private Sub Command1_Click()  
For i = 1 To 30  
Temp(i) = 0  
Next i  
For i = 1 To 30  
Temp(i) = Val(InputBox("Ввод элемента:", "Ввод массива"))  
Next i
```

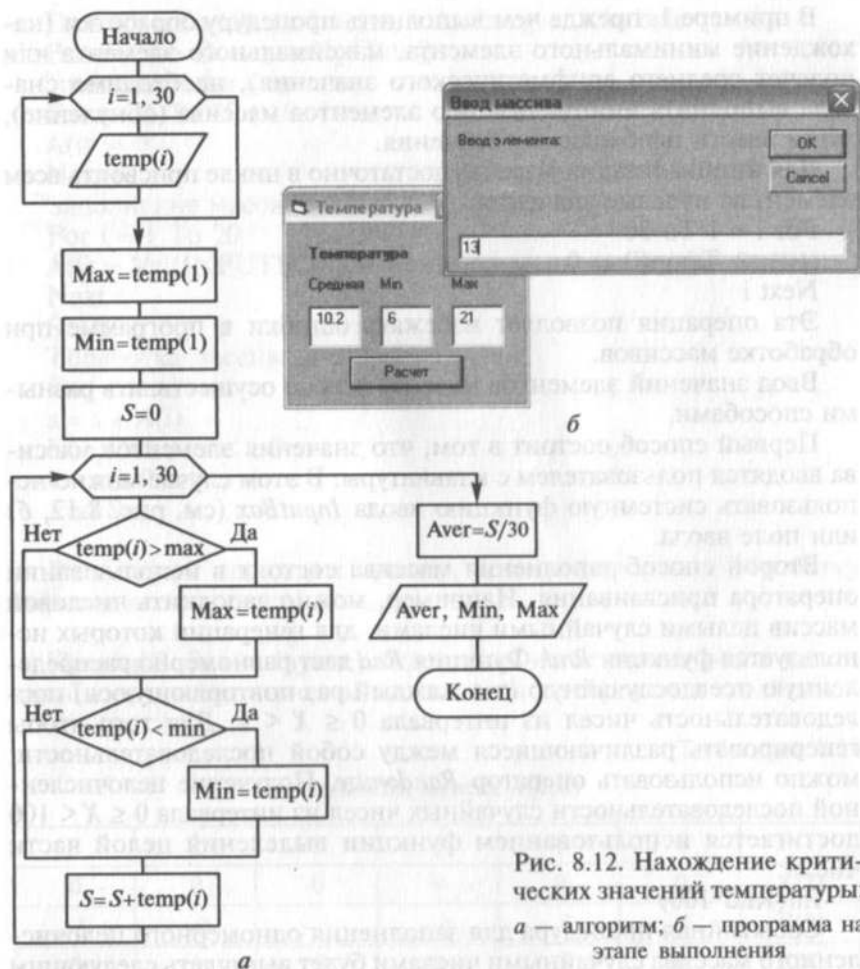


Рис. 8.12. Нахождение критических значений температуры:
а — алгоритм; б — программа на этапе выполнения

```

Max = Temp(1)
Min = Temp(1)
s = 0
For i = 1 To 30
  If Temp(i) > Max Then Max = Temp(i)
  If Temp(i) < Min Then Min = Temp(i)
  s = s + Temp(i)
Next i
Aver = s/30
Text1.Text = Str(Aver)
Text2.Text = Str(Min)
Text3.Text = Str(Max)
End Sub
  
```

В примере 1, прежде чем выполнить процедуру обработки (нахождение минимального элемента, максимального элемента или подсчет среднего арифметического значения), необходимо сначала выполнить инициализацию элементов массива (обнуление), затем ввести необходимые значения.

Для инициализации массива достаточно в цикле присвоить всем элементам нулевые значения:

```
For i = 1 To 30  
    Temp(i) = 0  
Next i
```

Эта операция позволяет избежать ошибки в программе при обработке массивов.

Ввод значений элементов массива можно осуществлять разными способами.

Первый способ состоит в том, что значения элементов массива вводятся пользователем с клавиатуры. В этом случае можно использовать системную функцию ввода *InputBox* (см. рис. 8.12, б) или поле ввода.

Второй способ заполнения массива состоит в использовании оператора присваивания. Например, можно заполнить числовой массив целыми случайными числами, для генерации которых используется функция *Rnd*. Функция *Rnd* дает равномерно распределенную псевдослучайную (т. е. каждый раз повторяющуюся) последовательность чисел из интервала $0 \leq X < 1$. Для того чтобы генерировать различающиеся между собой последовательности, можно использовать оператор *Randomize*. Получение целочисленной последовательности случайных чисел из интервала $0 \leq X < 100$ достигается использованием функции выделения целой части числа:

```
Int(Rnd*100)
```

Событийная процедура для заполнения одномерного целочисленного массива случайными числами будет выглядеть следующим образом:

```
Dim A(1 to 100) As Byte, i As Byte  
Private Sub Command1_Click()  
    For i=1 to 100  
        A(i)=Int(Rnd*100)  
    Next i  
End Sub
```

В большинстве случаев для работы с массивом используется цикл с параметром. Но не всегда он бывает удобен. Рассмотрим такой случай, когда обработка массива выполняется циклом с условием.

Пример 2. Подсчитать сумму элементов массива до первого нулевого элемента.

```

Dim A (1 To 20) As Integer, s As Integer, i As byte
Private Sub Command1_Click()
'инициализация массива
For i = 1 To 20
A(i) = 0
Next i
'заполнение массива
For i = 1 To 20
A(i) = Val(INPUTBOX("Введите элемент массива" + Str(i)))
Next i
S = 0
'обработка массива
Do
s = s + A(i)
Loop While x <> 0
Form1.Print
End Sub

```

Рассмотрим еще один пример, демонстрирующий обработку двумерного массива.

Пример 3. Задан двумерный массив из четырех строк и шести столбцов, подсчитать суммы элементов по строкам и столбцам.

Таблица 8.10

Исходный массив A(4,6)

1	2	3	4	5	6	S(1)
0	0	0	0	0	0	S(2)
-1	-2	-3	-4	-5	-6	S(3)
0	0	0	0	0	0	S(4)
B(1)	B(2)	B(3)	B(4)	B(5)	B(6)	

Из табл. 8.10 видно, что суммы элементов по строкам и столбцам можно представить как два одномерных массива S и B с числом элементов соответственно 4 и 6.

Задача разбивается на следующие подзадачи:

- инициализация и ввод элементов массива;
- обработка по строкам и столбцам;
- вывод полученных результатов.

Инициализация и ввод данных осуществляется с помощью вложенных циклов:

```

For i = 1 To 4
For j = 1 To 6

```

$A(i,j) = 0$

Next i

Next j

Последовательность обнуления элементов следующая: $A(1,1)$, $A(1,2)$, ..., $A(1,6)$; $A(2,1)$, ..., $A(2,6)$; $A(3,1)$, ..., $A(4,5)$, $A(4,6)$.

Аналогичным способом осуществляется и заполнение массива.

Для проверки ввода поставим задачу — вывести на экранную форму значения элементов массива A , как показано в табл. 8.10.

For i = 1 To 4

For j = 1 To 6

$S(i) = S(i) + A(i,j)$

Next i

Next j

For i = 1 To 4

For j = 1 To 6

$B(i) = B(i) + A(j,i)$

Next i

Next j

Очевидно, что перед началом обработки массивы S и B должны быть обнулены. Если матрица квадратная, то обработка будет производиться двумя циклами.

Предлагаем самостоятельно разработать вывод двух одномерных массивов, используя метод Print.

Использование массивов приводит к существенному уменьшению программного кода, т.е. к ее оптимизации.

Контрольные вопросы

1. В чем состоит различие между переменной и массивом?
2. Как происходит обращения к элементу массива?
3. Какие способы заполнения массива вы знаете?
4. Какой объем памяти будет отведен массиву $A(3 \text{ To } 20)$ типа Single?
5. Можно ли помещать в один массив фамилии и оценки студентов?
6. Дополните программу нахождения максимального и минимального значений температуры так, чтобы она фиксировала дни, соответствующие этим значениям.
7. Разработайте проект, в котором массив заполняется значениями текущего времени.
8. Разработайте проект, осуществляющий сортировку массива фамилий студентов.
9. Разработайте проект, моделирующий работу обменного пункта. В программе обеспечьте ввод курса купли-продажи валюты, которую обменный пункт покупает или продает.
10. Разработайте проект, который вычисляет стоимость продукта с учетом скидки. Скидка 3 % предоставляется в том случае, если сумма покупки больше 500 р.; в 5 % — если сумма покупки больше 1000 р.
11. Разработайте проект, который осуществляет вывод примера на умножение двух однозначных чисел, запрашивает ответ пользователя, про-

веряет его и выводит сообщения: «Правильно!» или «Вы ошиблись», а также правильный результат при необходимости.

12. Разработайте проект, который запрашивает у пользователя номер дня недели и выводит одно из сообщений: «Рабочий день», «Суббота» или «Воскресенье».

13. Разработайте проект, выводящий таблицу квадратов первых десяти целых положительных чисел.

14. Разработайте проект, который выводит таблицу степеней двойки (от нулевой до десятой).

8.6. ГРАФИЧЕСКИЕ МЕТОДЫ

В языке Visual Basic есть графические средства, к которым относятся графические методы и графические объекты. Эти средства наделают процесс проектирования Windows-приложений и сами эти приложения многими интересными и полезными возможностями.

Графический метод — это такой метод, который позволяет изображать на объекте данного класса какой-нибудь геометрический элемент, например точку, отрезок, прямоугольник, окружность и т.д.

Существуют следующие графические методы:

- метод *Scale*, с помощью которого вы можете задать на данном объекте подходящую для вашей задачи декартову систему координат;
- метод *PSet*, с помощью которого вы можете изобразить на данном объекте точку в заданной (с помощью метода *Scale* или по умолчанию) системе координат;
- метод *Line*, с помощью которого вы можете изобразить на данном объекте отрезок с заданными декартовыми координатами его концов, а также прямоугольник с заданными декартовыми координатами его углов;
- метод *Circle*, с помощью которого вы можете строить на данном объекте окружности, круги, эллипсы, дуги и секторы кругов и эллипсов с заданными координатами их центров, радиусами и другими характеристиками;
- метод *Cls*, с помощью которого данный объект полностью очищается от всего, что было построено на нем с использованием перечисленных графических методов или метода *Print*.

8.6.1. Графические методы *Scale* и *PSet*

Метод *Scale*. Этот метод применяется к объектам «Экранная форма» («Form») и «Графическое окно» («PictureBox»).

Объект «Графическое окно» — это элемент управления. Его можно создать с помощью инструмента, который входит в состав стандартного набора инструментов *ToolBox Visual Basic* .

Объекты «Form» и «PictureBox» имеют абсолютные координаты — положение на экране (объект «Form») или положение на экранной форме (объект «PictureBox»), а также размеры (ширину и высоту). Эти координаты измеряются в особых единицах — твипах.

Абсолютные координаты выражаются довольно большими числами, что очень неудобно. Поэтому для решения многих графических задач, например для рисования фигур или построения графиков, целесообразно задать новую систему координат. Эта система координат привязывается к тому объекту, на котором будет строиться изображение (говорят также, что объекту назначается новая система координат).

Часто, например, удобно считать, что начало координат находится посередине объекта, ось x направлена вправо, а ось y — вверх.

Метод Scale решает задачу назначения объекту новой системы координат. Синтаксис метода таков:

[ИмяОбъекта].Scale (X1, Y1) – (X2, Y2)

«ИмяОбъекта» может отсутствовать. В этом случае по умолчанию метод Scale выполняется для объекта *Экранная форма*.

(X1, Y1) — новые координаты левого верхнего угла объекта, а (X2, Y2) — новые координаты правого нижнего угла объекта.

Сразу после выполнения метода Scale старые (абсолютные) координаты объекта будут заменяться новыми.

Пример 1. Необходимо, чтобы объект класса «Графическое окно» с именем Picture1, который уже размещен на экранной форме, имел ширину 14 (в условных единицах), высоту 10 и чтобы координатные оси располагались привычным для вас образом: ось x была бы направлена слева направо, ось y — снизу вверх, а начало координат располагалось в центре данного объекта (рис. 8.13, а).

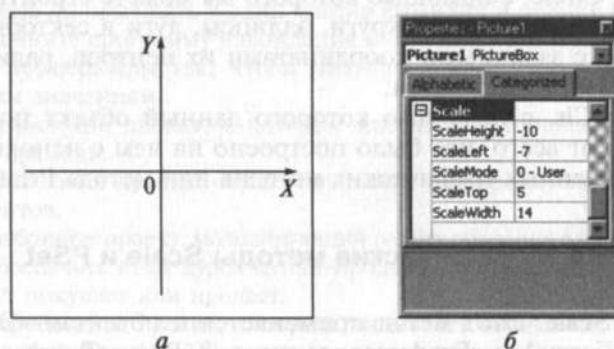


Рис. 8.13. Графическое окно:

а — расположение осей координат; б — установка значений свойств категории Scale

Сделать это можно с помощью следующей строки программного кода:

```
Picture1.Scale (-7, 5) - (7, -5)
```

В Visual Basic есть и другой способ назначения объекту новой системы координат.

Вместо применения метода Scale можно воспользоваться установкой значений свойств объекта из категории Scale (рис. 8.13, б).

Прежде всего следует установить значение свойства *ScaleMode* равным нулю (User).

Затем устанавливаются значения свойств *ScaleWidth* (*Ширина объекта* — разность координат правого и левого краев объекта), *ScaleHeight* (*Высота объекта* — разность координат правого и левого краев объекта), *ScaleLeft* (*Левый край* — координата левого края объекта) и *ScaleTop* (*Верхний край* — координата верхнего края объекта).

На рис. 8.14, б вы видите значения перечисленных свойств для примера 1. Обратите внимание на то, что значение свойства *ScaleHeight* — отрицательное число. Это следует из определения этого свойства как разности координат нижнего и верхнего краев объекта. В данном случае эта разность равна $(-5) - 5 = -10$. А вот значение свойства *ScaleWidth* положительно: $7 - (-7) = 14$.

Метод PSet. Этот метод применяется к тем же объектам, к которым применяется и метод Scale.

Синтаксис метода следующий:

```
[ИмяОбъекта.] PSet (X, Y) [, Цвет]
```

«ИмяОбъекта» может отсутствовать. В этом случае по умолчанию метод PSet используется *Экранной формой*.

(X, Y) — координаты «точки» (маленького цветного пятна на экране), которую вы хотите изобразить на *Объекте*.

«Цвет» — это выражение, значением которого является число типа Long. Этим числом в Visual Basic кодируется тот или иной цвет. Удобно в качестве цвета использовать константы Visual Basic: vbWhite, vbRed, vbGreen и т.д. (константы и числовые значения цветов приведены в табл. 8.11).

Если перед выполнением метода PSet метод Scale не применялся, то в качестве координат X и Y будут использоваться абсолютные координаты объекта.

Очевидно, что интерес представляет рисование не одной точки, а большого числа точек. Это необходимо при построении графиков функций.

Пример 2. С помощью метода PSet в графическом окне нарисуем участок звездного неба, например созвездие Большая Медведица.

Семь звезд этого созвездия — это семь черных «точек» на белом фоне объекта «Графическое окно», нарисованных с помощью метода PSet (рис. 8.14, а).

Таблица 8.11

Соответствие констант и числовых значений

Цвет	Константа	Числовое значение
Черный	vbBlack	0
Красный	vbRed	255
Зеленый	vbGreen	65 280
Желтый	vb Yellow	65 535
Синий	vbBlue	16 711 680
Сиреневый	vbMagenta	16 711 935
Голубой	vbCyan	16 776 960
Белый	vb White	16 777 215

Примечание. Цвет может отсутствовать. В этом случае по умолчанию его значение будет 0 — код черного цвета.

```
Private Sub Command1_Click()
Picture1.Scale (0, 10) — (15, 0)
Picture1.PSet (1, 6), vbBlack
Picture1.PSet (4, 8), vbBlack
Picture1.PSet (6.5, 6), vbBlack
Picture1.PSet (8.5, 8), vbBlack
Picture1.PSet (10, 2), vbBlack
Picture1.PSet (13, 3), vbBlack
Picture1.PSet (13, 7), vbBlack
End Sub
```

Изображения звезд — это, конечно, не точки, а довольно крупные пятна. Их размер можно легко изменить — для этого следует

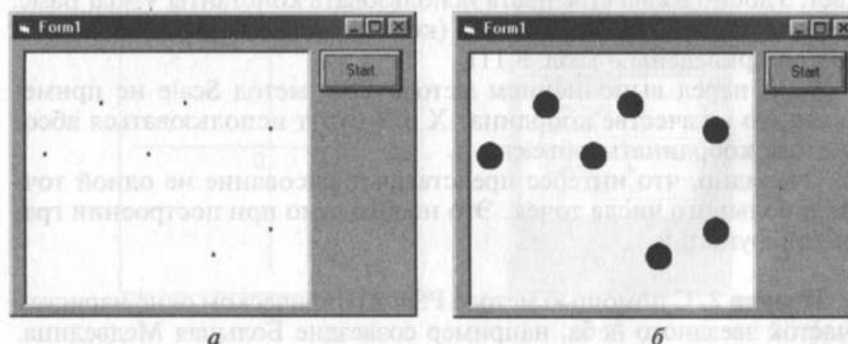


Рис. 8.14. Использование метода PSet:

a — размеры «точек», принятые по умолчанию; *б* — изменение размеров точек

установить нужное значение свойства *DrawWidth* объекта «Picture1». Значением свойства *DrawWidth* может быть целое число (1, 2, 3, ...). Для изображения на рис. 8.14, *а* это значение равно 2, а для изображения на рис. 8.14, *б* оно равно 20.

Изменить указанное значение можно либо с помощью окна *Properties*, либо вставив в программный код строку
`Picture1.DrawWidth = 20`

8.6.2. Установка цвета

По умолчанию все методы, используемые для изображения на объекте геометрических элементов, осуществляют рисование черным цветом.

У объекта, на котором производится рисование, есть свойства, которые определяют внешний вид изображаемой фигуры:

- *ForeColor* (цвет контура);
- *FillStyle* (стиль заполнения);
- *FillColor* (цвет заполнения).

В языке Visual Basic цвет задается либо с помощью длинного целого числа, либо с помощью одной из восьми констант. Значения этих констант можно представить в виде суммы трех слагаемых: $R + 256G + 256^2B$, где каждая из переменных (*R*, *G* и *B*) принимает одно из двух значений: 0 или 255.

Переменные *R*, *G* и *B* могут принимать и другие целочисленные значения от 0 до 255, например: $R = 64$, $G = 128$ и $B = 192$ (табл. 8.12). При этом указанная сумма будет отличаться от значений, представленных в табл. 8.11, а обозначаемый этой суммой цвет будет отличаться от цветов, приведенных в табл. 8.12.

Таблица 8.12

Значения функции *RGB*

Цвет	Интенсивность красного	Интенсивность зеленого	Интенсивность синего
Черный	0	0	0
Белый	255	255	255
Красный	255	0	0
Зеленый	0	255	0
Синий	0	0	255
Бирюзовый	0	255	255
Пурпурный	255	0	255
Желтый	255	255	0
Серый	192	192	192

В языке Visual Basic есть функция *RGB (R, G, B)*, значение которой — это значение указанной суммы, имеющее тип Long (длинное целое число).

Очевидно, что число значений, которые может принимать функция *RGB*, равно 256^3 . Это очень большое число — именно такое число различных цветовых оттенков могут использовать графические методы Visual Basic.

А как по значениям *R*, *G* и *B* можно определить, что это за оттенки?

Ответ на этот вопрос прост. Каждый оттенок представляет собой смесь всего трех цветов: Red (красного), Green (зеленого) и Blue (синего). Значение переменной *R* — это доля красного цвета в этой смеси, значение переменной *G* — доля зеленого цвета, а значение переменной *B* — доля синего цвета.

Если все три значения одинаковы, цвета нейтрализуют друг друга — получается черный, серый или белый цвет (чем меньше это значение, тем цвет ближе к черному; чем больше это значение, тем цвет ближе к белому).

Если значения *R*, *G* и *B* не совпадают, то получаются оттенки самых разных цветов. Например, если $R=0$, а $G=B$, то получается голубой цвет (Cyan); если $G=0$, а $R=B$, то получается сиреневый цвет (Magenta); если же $B=0$, а $R=G$, получается желтый цвет (Yellow). Все эти цвета будут более яркими, если ненулевые значения ближе к 255, и будут приближаться к черному, если ненулевые значения приближаются к 0.

Для задания цвета контура и цвета заполнения необходимо присвоить свойствам *ForeColor* и *FillColor* целочисленное значение (1...8) или одну из восьми констант, например:

Form1.ForeColor = vbCyan

Form1.FillColor = vbRed

Таблица 8.13

Константы функции QBColor

Цвет	Константа	Цвет	Константа
Черный (vbBlack)	0	Темно-серый	8
Темно-синий	1	Синий (vbBlue)	9
Темно-зеленый	2	Зеленый (vbGreen)	10
Темно-голубой	3	Голубой (vbCyan)	11
Темно-красный	4	Красный (vbRed)	12
Темно-сиреневый	5	Сиреневый (vbMagenta)	13
Коричневый	6	Желтый (vbYellow)	14
Светло-серый	7	Белый (vbWhite)	15

Таблица 8.14

Константы заполнения

Константа	Вид штриховки	Константа	Вид штриховки
0	Solid	4	Upward Diagonal
1	Transparent	5	Downward Diagonal
2	Horizontal Line	6	Cross
3	Vertical Line	7	Diagonal Cross

Кроме перечисленных способов задания цвета в языке Visual Basic есть еще один способ — с помощью функции *QBColor*, единственным аргументом которой является целое число *C*, а значением — код цвета (число типа Long).

Названия этих цветов приведены в табл. 8.13 (в скобках указаны константы Visual Basic, которым соответствуют некоторые из них).

При задании цвета контура или цвета заполнения можно использовать функцию *QBColor*, например:

```
Picture1.ForeColor = QBColor(3)
```

```
Picture1.FillColor = QBColor(8)
```

Для задания стиля заполнения необходимо выполнить следующую команду:

```
Picture1.FillStyle = 2
```

В табл. 8.14 приведен список значений свойства *FillStyle*, которые обеспечивают автоматическое заполнение фигур различными видами штриховки.

8.6.3. Использование графического метода Line и метода Print

Метод *Line* широко используется в программировании самых разнообразных задач. Метод применяется к объектам «Экранная форма» («Form») и «Графическое окно» («PictureBox»).

Метод *Line* позволяет:

- чертить на объекте, к которому этот метод применяется, отрезки прямых линий;
- изображать на этом объекте закрашенные или незакрашенные прямоугольники.

Метод *Line* имеет следующий синтаксис:

```
[ИмяОбъекта]. Line (X1, Y1) – (X2, Y2) [, Цвет [, Флаг ]]
```

«ИмяОбъекта» может отсутствовать. В этом случае по умолчанию метод применяется к экранной форме.

Вслед за ключевым словом «Line» через запятую следуют параметры метода.

Параметр (X1, Y1) – (X2, Y2) — это координаты границ отрезка или двух углов прямоугольника — левого верхнего и правого нижнего (или, наоборот, правого нижнего и левого верхнего).

Параметр «Цвет» аналогичен параметру метода PSet.

Параметр «Цвет» может отсутствовать. Тогда по умолчанию он будет совпадать со значением свойства *ForeColor* объекта.

Параметр «Флаг» — это либо символ «B», либо символы «BF». В первом случае рисуется не отрезок, а прямоугольник, во втором — закрашенный прямоугольник. (Цвет закрашки — это значение параметра «Цвет».)

Если в качестве флага используется символ «B», то прямоугольник ничем не будет заполнен, тогда у объекта, в котором он рисуется, значение свойства *FillStyle* равно 1 (Transparent). Именно это значение установлено у этого свойства по умолчанию. Если указанное свойство имеет значение 0 (Solid), то прямоугольник будет сплошь заполнен («залит») цветом, который является значением свойства *FillColor*.

Если «Флаг» есть, а «Цвет» отсутствует, то «лишнюю» запятую отбрасывать нельзя, так как «Флаг» будет воспринят системой как «Цвет».

Толщину отрезка или контура прямоугольника можно увеличить. Для этого, как и в методе PSet, следует увеличить значение свойства *DrawWidth* (по умолчанию оно равно 1).

Пример 3. Нарисуем на экранной форме две прямые разного цвета и разной толщины, а также два прямоугольника, один из которых залит цветом контура. На рис. 8.15, а приведена экранная форма с демонстрацией использования метода Line.

Процедура-событие, вызывающая рисование представленных на рис. 8.15, а фигур, следующая:

```
Private Sub Command1_Click()
```

```
'Изображение тонкого красного горизонтального отрезка:
```

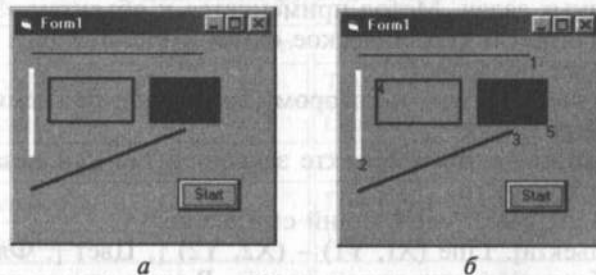


Рис. 8.15. Рисование геометрических фигур и вывод текста:

а — рисование с помощью метода Line; б — совместное использование методов Line и Print

```

Line (200, 200)–(2200, 200), 255
'Изображение толстого голубого вертикального отрезка:
DrawWidth = 4
Line (200, 400)–(200, 1400), vbCyan
'Изображение красного наклонного отрезка средней толщины:
DrawWidth = 2
Line (200, 1800)–(2000, 1100), vbRed
'Изображение незакрашенного прямоугольника (цвет контура
черный — берется по умолчанию):
Line (1400, 1000)–(400, 500), , B
'Изображение красного закрашенного прямоугольника:
Line (1600, 500)–(2400, 1000), 255, BF
End Sub

```

Обратите внимание на то, что перед названием метода `Line` и перед названием свойства `DrawWidth` не написано имя объекта — это означает, что по умолчанию в качестве объекта используется *Экранная форма*. Но не было бы ошибкой, если бы имя экранной формы перед указанными словами было написано. Например: `Form1.DrawWidth = 2`.

Обратите внимание и на то, что в данном примере масштабирования экранной формы с помощью метода `Scale` не производилось, поэтому у прямоугольников и отрезков значения координат (абсолютных) — довольно большие числа.

В подразд. 8.3 рассмотрен метод `Print`, который используется для выдачи чисел и строк на *Объект*.

С помощью этого метода можно выводить данные и на экранную форму, и в графическое окно.

Метод `Print` может использоваться совместно с методом `Line` и другими графическими методами. При этом строки и числа будут печататься в том месте экранной формы или графического окна, где закончилось рисование отрезка или прямоугольника — чуть ниже и правее точки, на которой закончилось рисование.

Пример 4. После каждой строки кода примера 3, содержащей метод `Line`, поместим строку с печатью целого числа — от 1 до 5. Результат работы этой программы показан на рис. 8.15, б. Процедура-событие будет выглядеть следующим образом:

```

Private Sub Command1_Click()
'Изображение тонкого красного горизонтального отрезка:
Line (200, 200) – (2200, 200), 255
Print «1»
'Изображение толстого голубого вертикального отрезка:
DrawWidth = 4
Line (200, 400) – (200, 1400), vbCyan
Print «2»
'Изображение красного наклонного отрезка средней толщины:

```

```

DrawWidth = 2
Line (200, 1800) – (2000, 1100), vbRed
Print «3»
'Изображение незакрашенного прямоугольника (цвет контура
черный — берется по умолчанию):
Line (1400, 1000) – (400, 500), , B
Print «4»
'Изображение красного закрашенного прямоугольника:
Line (1600, 500) – (2400, 1000), 255, BF
Print «5»
End Sub

```

8.6.4. Метод Circle

Этот метод применяется к тем же объектам, к которым применяются и предыдущие графические методы.

Синтаксис метода следующий:

[ИмяОбъекта]. Circle (X, Y), Радиус [, Цвет [, Угол1 , Угол2 [, КоэффициентСжатия]]]

«ИмяОбъекта» может отсутствовать. В этом случае по умолчанию метод Circle используется *Экранной формой*.

(X, Y) — координаты центра круга или эллипса, который (или часть которого) будет создавать метод Circle на *Объекте*.

«Цвет» — это выражение, которое определяется так же, как и для метода Line.

«Угол1» и «Угол2» — это начальный и конечный углы дуги или сектора.

Углы измеряют в радианах, их значения берутся в интервале от 0 до 2π . Нулевой угол соответствует горизонтальной оси, направленной вправо.

Если перед ненулевым значением угла ставится знак «-», то это означает, что будет нарисована не дуга, а сектор.

При рисовании дуги или сектора движение «пера» происходит всегда против часовой стрелки, от «Угла1» к «Углу2» (и когда «Угол2» больше «Угла1», и когда «Угол1» больше «Угла2»).

Если изображается не дуга и не сектор круга или эллипса, а полный круг или эллипс, то углы, естественно, отсутствуют.

«Коэффициент Сжатия» — это положительное число, большее или меньшее единицы. В первом случае получается эллипс, вытянутый по вертикали, а во втором — эллипс, вытянутый по горизонтали. Степень вытянутости определяется значением коэффициента. Если он отсутствует, то по умолчанию изображается круг (окружность).

Как и в методе Line, не только значения перечисленных параметров метода Circle влияют на вид фигуры, которая будет пост-

роена этим методом. Напомним, что у объекта, на котором производится рисование, есть свойства, которые определяют внешний вид изображаемой фигуры: цвет ее контура (свойство *ForeColor*), стиль ее заполнения (свойство *FillStyle*), а также цвет ее заполнения (свойство *FillColor*).

Приведем пример программы, которая демонстрирует метод *Circle* и свойств объектов, для которых этот метод применяется (рис. 8.16).

Обратите внимание на то, что для равномерной заливки фигуры каким-нибудь цветом свойство *FillStyle* Объекта должно иметь значение 0, а непосредственно перед вычерчиванием фигуры свойство *FillColor* должно получить значение, которое соответствует цвету заливки.

```
Событийная процедура:  
Private Sub Command1_Click()  
Picture1.Scale (0, 35) - (15, 0)  
Picture1.Circle (5, 25), 2.5, 0  
Picture1.ForeColor = vbCyan  
Picture1.FillStyle = 5  
Picture1.FillColor = vbRed  
Picture1.Circle (8, 6), 5.5, , , 0.5  
Picture1.ForeColor = vbBlue  
Picture1.FillStyle = 0  
End Sub
```

В программах часто бывает необходимо выполнить очистку экрана. С помощью метода *Cls* можно очистить форму или объект «picture» от текста и графики, созданных в нем программно. Синтаксис метода *Cls* очень простой и выглядит так:

```
object.Cls,
```

где «object» — объект, в котором применяется метод. Если параметр не указан, то по умолчанию используется форма.

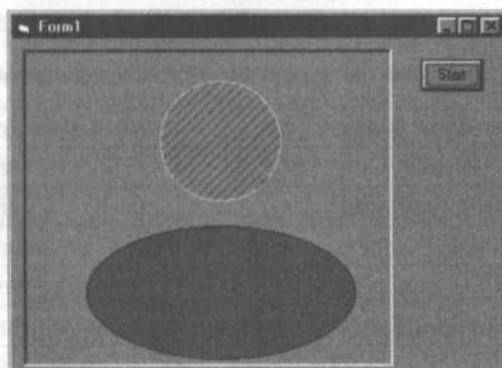


Рис. 8.16. Демонстрация метода *Circle*

Необходимо помнить, что при применении этого метода свойства *CurrentX* и *CurrentY* выбранного объекта устанавливаются в 0.

8.6.5. Работа с изображениями

Для размещения на форме графических изображений используются объекты «*image*» и «*picture*». Однако в качестве изображений, помещаемых в эти объекты, можно использовать только файлы определенных типов. Допустимые типы файлов изображений представлены в табл. 8.15.

Изображение размещается в объекте «*image*» и на форме, т.е. при помощи свойства *picture*. Однако, в отличие от размещения изображения непосредственно на форме, изображение в объекте «*image*» пропорционально изменяет свои размеры. Изменением размера управляет свойство *Stretch*. Если это свойство имеет значение «*False*», то объект «*image*» принимает размеры в соответствии с размером изображения. Если оно имеет значение равно «*True*», то размеры изображения изменяются в соответствии с размерами объекта.

Изображение в объекте *Picture* загружается полностью. При этом если оно больше размеров объекта, то видна только часть изображения; если меньше, — то изображение размещается в левом верхнем углу. Если установить для свойства *AutoSize* объекта *Picture* значение «*True*», то изображение будет пропорционально вписываться в объект *Picture*.

Определить графическое изображение в приложении можно как в режиме проектирования (режим *Design Time*), так и в режиме выполнения (режим *Run Time*). Чтобы задать графическое изображение в режиме проектирования, используется свойство *Picture* объекта. Для этого необходимо выполнить следующие действия.

Таблица 8.15

Допустимые типы файлов изображений для графических объектов

Описание файла	Тип файла (расширение)
Битовый растровый файл	bmp
Независимый растровый файл	dib
Файл значка	ico,cur
Метафайл	wmf
Расширенный метафайл	emf
Растровый файл	gif
Растровый сжатый файл	jpg, Jpeg

1. Открыть окно *Properties* объекта.
2. Выбрать свойство *Picture*.
3. Нажать кнопку с тремя точками в правом столбце свойства.
4. В открывшемся диалоговом окне *LoadPicture* найти требуемый файл.
5. Дважды щелкнуть на файле мышью или установить на него курсор и нажать кнопку *Открыть*.

В режиме выполнения программы существует больше возможностей для определения изображения объекта. Изображение можно задать следующими способами:

- загрузить изображение при помощи функции *LoadPicture* с указанием файла изображения. При этом изображение помещается в свойство *Picture* объекта. Например:

```
Object.Picture = LoadPicture("C:\...\Картинка.bmp");
```

- загрузить изображение из ресурсного файла приложения при помощи функции *LoadResPicture* с указанием идентификатора ресурса и его типа:

```
Set Object.Picture = LoadResPicture(Resident, vbResBMP);
```

- скопировать изображение из одного объекта в другой с помощью обычного присвоения:

```
ObjectTo.Picture = ObjectFrom.Picture;
```

Удаляется изображение из объекта очень просто. Для этого необходимо воспользоваться загрузкой в свойство *Picture* пустого значения:

```
Set Объект.Picture = LoadPicture(" ")
```

При этом изображение удаляется из объекта.

Контрольные вопросы

1. Почему необходимо задание новой системы координат для решения графических задач?
2. К каким объектам применяются графические методы?
3. Если при вызове графического метода отсутствует имя объекта, то к какому объекту он применяется?
4. Каким образом можно осуществить движение графического объекта?
5. Используя графический метод, нарисуйте дачный участок.
6. Разработайте приложение «Крути на воде».
7. Разработайте проект, в котором по выбору фамилии студента выводится его фотография.
8. Разработайте программу, которая вычерчивает на экране домик.
9. Разработайте приложение, которое выводит на экран флаг Олимпийских игр (разноцветный).
10. Разработайте программу, которая вычерчивает на экране кораблик.
11. Разработайте приложение, которое выводит на экран изображение шахматной доски.
12. Разработайте проект, который осуществляет движение окружности по экрану.

ГЛОССАРИЙ

Адаптер — устройство связи компьютера с периферийными устройствами.

Адрес — номер конкретного байта оперативной памяти компьютера.

Алгебра логики — раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Алгоритм — заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Алфавит — фиксированный для данного языка набор основных символов, т. е. букв алфавита, из которых должен состоять любой текст на этом языке. Никакие другие символы в тексте не допускаются.

Антивирусные программы — программы, предотвращающие заражение компьютерным вирусом и ликвидирующие последствия заражения.

Арифметико-логическое устройство (АЛУ) — часть процессора, которая производит выполнение операций, предусмотренных данным компьютером.

Архитектура компьютера — логическая организация, структура и ресурсы компьютера, которые может использовать программист. Определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера.

Архитектура фон Неймана — архитектура компьютера, имеющего одно арифметико-логическое устройство, через которое проходит поток данных, и одно устройство управления, через которое проходит поток команд.

Ассемблер — см. Язык ассемблера.

Аудиоадаптер (звуковая плата) — специальная электронная плата, которая позволяет записывать звук, воспроизводить и создавать его программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

База данных — один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

Байт — группа из 8 бит, рассматриваемая при хранении данных как единое целое.

Библиотека стандартных подпрограмм — совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм.

Бит — наименьшая единица информации в цифровом компьютере, принимающая значение 0 или 1.

Ввод — считывание информации внешнего устройства в память компьютера.

Вещественное число — в информатике тип данных, содержащий числа, записанные с десятичной точкой и (или) с десятичным порядком.

Видеоадаптер — электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит видеопамять, регистры ввода-вывода и модуль BIOS. Посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

Винчестер — см. **Накопитель на жестких магнитных дисках**.

Вирус компьютерный — специально написанная небольшая программа, которая может «приписывать» себя к другим программам для выполнения каких-либо вредных действий: портит файлы, «засоряет» оперативную память.

Внешняя память — совокупность запоминающих устройств для длительного хранения данных. В ее состав входят накопители на гибких и жестких магнитных дисках, оптические и магнитооптические накопители, накопители на магнитной ленте. Во внешней памяти обычно хранятся архивы программ и данных. Информация, размещенная на внешних носителях, не зависит от того, включен или выключен компьютер.

Вывод — результаты работы программы, выдаваемые компьютером пользователю, другому компьютеру или во внешнюю память.

Выражение — в языке программирования — запись правила для вычисления некоторого значения. Строится из констант, переменных и указателей функций, объединенных знаками операций.

Гибкий (флоппи-) диск — круглая пластинчатая пластина, покрытая с обеих сторон магнитным окислом и помещенная в защитную оболочку. Используется как носитель небольших объемов информации.

Графический редактор — программа, позволяющая создавать и редактировать изображения на экране компьютера: рисовать линии, раскрашивать области экрана, создавать надписи различными шрифтами, обрабатывать изображения, полученные с помощью сканеров. Некоторые редакторы обеспечивают возможность получения изображений трехмерных объектов, их сечений и разворотов.

Графопостроитель — устройство для вывода из компьютера информации в виде графиков и чертежей на неподвижную или вращающуюся на барабане бумагу.

Джойстик — стержень-ручка, отклонение которой от вертикального положения приводит к передвижению курсора в соответствующем направлении по экрану дисплея. Часто применяется в компьютерных играх.

Диск — круглая металлическая или пластмассовая пластина, покрытая магнитным материалом, на которую информация наносится в виде концентрических дорожек, разделенных на секторы.

Дисковод — устройство, управляющее вращением магнитного диска, чтением и записью данных на нем.

Дисплей — устройство визуального отображения информации (в виде текста, таблицы, рисунка, чертежа и др.) на экране электронно-лучевого прибора.

Драйверы — программы, расширяющие возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д. С помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся устройств.

Идентификатор — символическое имя переменной, которое идентифицирует ее в программе.

Инструментальные программные средства — программы, используемые в ходе разработки, корректировки или развития других программ: редакторы, отладчики, вспомогательные системные программы, графические пакеты и др. По назначению близки системам программирования.

Интегральная схема — реализация электронной схемы, выполняющей некоторую функцию, в виде единого полупроводникового кристалла, в котором изготовлены все компоненты, необходимые для осуществления этой функции.

Интегрированные пакеты программ — пакеты программ, выполняющие ряд функций, для которых ранее создавались специализированные программы, в частности текстовые редакторы, электронные таблицы, системы управления базами данных, программы построения графиков и диаграмм.

Интернет — гигантская всемирная компьютерная сеть, объединяющая десятки тысяч сетей всего мира. Ее назначение — обеспечить любому желающему постоянный доступ к любой информации. Интернет предлагает практически неограниченные информационные ресурсы, полезные сведения, учебу, развлечения, возможность общения с компетентными людьми, услуги удаленного доступа, передачи файлов, электронной почты и многое другое. Интернет обеспечивает принципиально новый способ общения людей, не имеющий аналогов в мире.

Интерпретатор — разновидность транслятора. Переводит и выполняет программу с языка высокого уровня в машинный код строка за строкой.

Интерфейс — электронная схема сопряжения двух устройств, обменивающихся информацией.

Информатизация общества — организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов.

Информатика — дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы ее создания, хранения, поиска, преобразования, передачи и использования в различных сферах человеческой деятельности. За понятием «информатика» закреплены области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты компьютеризации — массового внедрения компьютерной техники во все области жизни людей. Информатика в самом своем существе базируется на компьютерной технике.

Информационная система — взаимосвязанная совокупность средств, методов и персонала, участвующих в обработке данных.

Информационная технология — совокупность методов, производственных и программно-технических средств, объединенных в технологическую цепочку, обеспечивающую сбор, хранение, обработку, вывод и распространение информации для снижения трудоемкости процессов использования информационных ресурсов, повышения надежности и оперативности.

Информационно-поисковая система (ИПС) — система, выполняющая функции хранения большого объема информации, быстрого поиска требуемой информации, добавления, удаления и изменения хранимой информации, вывода ее в удобном для человека виде.

Информационный процесс — процесс, в результате которого осуществляется прием, передача (обмен), преобразование и использование информации.

Информация — сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний. Применительно к обработке данных на компьютерах — произвольная последовательность символов, несущих смысловую нагрузку.

Искусственный интеллект (ИИ) — дисциплина, изучающая возможность создания программ для решения задач, которые требуют определенных интеллектуальных усилий при выполнении их человеком. Примерами областей использования ИИ являются: игры, логический вывод, обучение, понимание естественных языков, формирование планов, понимание речи, доказательство теорем и визуальное восприятие.

Исполнитель алгоритма — человек или автомат (в частности, процессор компьютера), умеющий выполнять определенный набор действий. Исполнителя характеризуют среда, элементарные действия, система команд, отказы.

Итерационный цикл — вид цикла, для которого число повторений операторов тела цикла заранее неизвестно. На каждом шаге вычислений происходит последовательное приближение и проверка условия достижения искомого результата. Выход из цикла осуществляется в случае выполнения заданного условия.

Каталог (директория, папка) — оглавление файлов. Доступен пользователю через командный язык операционной системы. Его можно просматривать, переименовывать зарегистрированные в нем файлы, переносить их содержимое на новое место и удалять. Часто имеет иерархическую структуру.

Клавиатура компьютера — устройство для ввода информации в компьютер и подачи управляющих сигналов. Содержит стандартный набор клавиш печатающей машинки и некоторые дополнительные клавиши: управляющую клавишу, функциональные клавиши, клавиши управления курсором и малую цифровую клавиатуру.

Клиент (рабочая станция) — любой компьютер, имеющий доступ к услугам сервера. Клиентом также называют прикладную программу, которая от имени пользователя получает услуги сервера. См. также Сервер.

Клиент-сервер — модель взаимодействия пользователя с рабочими станциями, при которой сервер сначала обрабатывает запросы, а затем посылает клиенту то, в чем он нуждается.

Ключевое слово — слово языка программирования, имеющее определенный смысл для транслятора. Его нельзя использовать для других целей, например в качестве имени переменной.

Команда — описание элементарной операции, которую должен выполнить компьютер. Обычно содержит код выполняемой операции, указания по определенным операндам (или их адресам), указания по размещению получаемого результата. Последовательность команд образует программу.

Компакт-диск (CD-ROM) — постоянное запоминающее устройство, выполненное с использованием специальной оптической технологии. В ряду запоминающих устройств занимает место между флоппи- и жесткими дисками, являясь одновременно мобильным и очень емким.

Компилятор — транслятор, выполняющий преобразование программы, составленной на исходном языке, в объектный модуль.

Компьютер — программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами. Основу компьютеров образует аппаратура (Hardware), построенная в основном с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (Software) — заранее заданных, четко определенных последовательностей арифметических, логических и других операций.

Компьютеризация — задачи массового внедрения компьютеров во все области жизни, стоящие перед странами как необходимое, важное условие их прогресса и развития, а также последствия, которые будут вызваны массовым внедрением компьютеров. Цель компьютеризации — улучшение качества жизни людей за счет увеличения производительности и облегчения условий их труда.

Контроллер — устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

Курсор — светящийся символ на экране дисплея, указывающий позицию, на которой будет отображаться следующий вводимый с клавиатуры знак.

Кэш — см. **Сверхоперативная память**.

Логический тип — тип данных, представляемых значениями «истина» или «ложь» («да» или «нет»). Иногда также называется булевским в честь английского математика XIX в. Дж. Буля.

Логический элемент (вентиль) — часть электронной логической схемы, выполняющая элементарную логическую функцию.

Логическое высказывание — любое предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Локальная сеть — см. **Сеть компьютерная**.

Манипуляторы (джойстик, мышь, трекбол и др.) — специальные устройства для управления курсором.

Маршрутизатор — программа, определяющая путь передачи сообщения между узлами и абонентами сети.

Массив — последовательность однотипных элементов, число которых фиксировано и которым присвоено одно имя. Компьютерный эквивалент таблицы. Положение элемента в массиве однозначно определяется его индексами.

Математическая модель — система математических соотношений — формул, уравнений, неравенств и т. д., отражающих существенные свойства объекта.

Машинный язык — совокупность машинных команд компьютера, отличающаяся числом адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина, и др.

Мейнфрейм — большая ЭВМ с объемом оперативной памяти от 512 Мбайт до 8 Гбайт.

Меню — выведенный на экран дисплея список различных вариантов работы компьютера, по которому можно сделать конкретный выбор.

Метод — программа действий над объектом или его свойствами.

Микрокомпьютер — компьютер, в котором в качестве управляющего и арифметического устройства используется микропроцессор.

Микропроцессор — процессор, выполненный в виде интегральной схемы. Состоит из цепей управления, регистров, сумматоров, счетчиков команд, очень быстрой памяти малого объема.

Модем — устройство, обеспечивающее преобразование цифровых сигналов компьютера в переменный ток частоты звукового диапазона (модуляцию), а также обратное преобразование (демодуляцию). Используется для соединения компьютера с другими компьютерными системами через телефонную сеть.

Монитор — см. Дисплей.

Мост — устройство для соединения двух сетей, присутствие которого для абонентов невидимо (в отличие от шлюза).

Мультимедиа — собирательное понятие для различных компьютерных технологий, при которых используется несколько информационных сред, таких как графика, текст, видео, фотография, движущиеся образы (анимация), звуковые эффекты, высококачественное звуковое сопровождение. Мультимедиа-компьютер — это компьютер, снабженный аппаратными и программными средствами, реализующими технологию мультимедиа.

Мышь — устройство управления курсором. Имеет вид небольшой коробки, помещающейся на ладони. Связана с компьютером кабелем. Ее движения трансформируются в перемещения курсора по экрану дисплея.

Накопитель на жестких магнитных дисках (винчестерский накопитель) — наиболее массовое запоминающее устройство большой емкости, в котором носителями информации являются круглые алюминиевые пластины — платтеры, обе поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения больших объемов информации.

Ноутбук (блокнот) — портативный компьютер, по своим размерам близкий к книге крупного формата. Помещается в портфель-дипломат.

Обычно комплектуется модемом (для связи с офисом) и снабжается приводом CD-ROM.

Оболочки — программы, создаваемые для упрощения работы со сложными программными системами, такими как операционная система DOS. Они преобразуют неудобный командный пользовательский интерфейс в дружелюбный графический интерфейс или интерфейс типа меню. Оболочки предоставляют пользователю удобный доступ к файлам и обширные сервисные услуги.

Обработка информации — в информатике — любое преобразование информации из одного вида в другой, производимое по строгим формальным правилам.

Общая шина — топология сети, при которой ее узлы подключены к общему линейному информационному каналу.

Объект — реально существующий предмет в окружающем нас мире, обладающий состоянием, поведением и индивидуальностью. Состояние объекта определяется перечнем всех возможных свойств и текущим значением каждого из них.

Оперативная память — быстрое запоминающее устройство не очень большого объема, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Оператор — фраза алгоритмического языка, определяющая некоторый законченный этап обработки данных. В состав операторов входят ключевые слова, данные, выражения и др.

Операционная система — комплекс взаимосвязанных программ, предназначенных для автоматизации планирования и организации процесса обработки программ, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программ, других вспомогательных операций обслуживания. Важнейшая часть программного обеспечения.

Описание — раздел программы, идентифицирующий структуры данных, которыми должна манипулировать программа, и описывающий их типы.

Основание системы счисления — количество различных цифр, используемых для изображения чисел в данной системе счисления.

Отладка — этап компьютерного решения задачи, при котором происходит устранение явных ошибок в программе. Часто производится с использованием специальных программных средств — отладчиков.

Отладчик — программа, позволяющая исследовать внутреннее поведение разрабатываемой программы. Обеспечивает пошаговое исполнение программы с остановкой после каждой оператора, просмотр текущего значения переменной, нахождение значения любого выражения и др.

Пакет — часть сообщения, удовлетворяющая некоторому стандарту, последовательность битов, содержащая данные и сигналы управления, передаваемые и коммутируемая как одно целое.

Пакеты прикладных программ (ППП) — специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

Переменная — величина, значение которой может меняться в процессе выполнения программы.

Персональный компьютер — микрокомпьютер универсального назначения, рассчитанный на одного пользователя и управляемый одним человеком.

Пиктограмма — условное изображение информационного объекта или операции.

Платформа — тип процессора и ОС, на которых можно установить новый программный продукт.

Подпрограмма — самостоятельная часть программы, которая создается независимо от других частей и затем вызывается по имени. Когда имя подпрограммы используется в качестве оператора программы, выполняется вся группа операторов, представляющая тело подпрограммы.

Поколения компьютеров — условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с ними.

Порты устройств — электронные схемы, содержащие один или несколько регистров ввода-вывода и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора. Последовательный порт обменивается данными с процессором побайтно, а с внешними устройствами — побитно. Параллельный порт получает и посылает данные побайтно.

Постоянная память — энергонезависимое постоянное запоминающее устройство (ПЗУ), изготовленное в виде микросхемы. Используется для хранения данных, не требующих изменения. Содержание памяти специальным образом «зашивается» в ПЗУ при изготовлении. В ПЗУ находятся программа управления работой самого процессора, программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Почтовый ящик — специально организованный файл для хранения корреспонденции.

Прикладная программа — любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

Приложение — совокупность программ, реализующих обработку данных в определенной области применения.

Принтер — печатающее устройство. Преобразует закодированную информацию, выходящую из процессора, в форму, удобную для чтения на бумаге.

Принцип открытой архитектуры.

1. Регламентируются и стандартизируются только описание принципа действия компьютера и его конфигурация (определенная совокупность аппаратных средств и соединений между ними). Таким образом, компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.

2. Компьютер легко расширяется и модернизируется за счет наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие

заданному стандарту, и тем самым устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.

Принципы фон Неймана.

1. Принцип программного управления. Программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

2. Принцип адресности. Основная память состоит из перенумерованных ячеек, процессору в произвольный момент времени доступна любая ячейка.

3. Принцип однородности памяти. Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Приоритет — число, приписанное задаче, процессу или операции и определяющее очередность их выполнения или обслуживания.

Программное обеспечение — совокупность программ, выполняемых компьютером, а также вся область деятельности по проектированию и разработке программ.

Прокрутка — имитация программистом за столом выполнения программы на конкретном наборе тестовых данных.

Протокол коммуникации — согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных. Имеются протоколы для скорости передачи, форматов данных, контроля ошибок и др.

Процесс — функция обработки данных любого вида на компьютере.

Псевдокод — система обозначений и правил, предназначенная для единообразной записи алгоритмов. Занимает промежуточное место между естественным и формальным языками.

Разделение времени — режим работы, при котором процессорное время предоставляется различным приложениям последовательно квантами. По истечении кванта времени приложение возвращается в очередь ожидания обслуживания.

Растр — множество точечных элементов, с помощью которых знак фиксируется на бумажном носителе или отображается на экране дисплея.

Реальное время — режим обработки данных, при котором обеспечивается взаимодействие вычислительной системы с внешними по отношению к ней процессами в темпе, соизмеримом со способностью протекания этих процессов.

Регистр — специальная запоминающая ячейка, выполняющая функции кратковременного хранения числа или команды и выполнения над ними некоторых операций. Отличается от ячейки памяти тем, что может не только хранить двоичный код, но и преобразовывать его.

Регистр команд — регистр устройства управления для хранения кода команды на период времени, необходимый для ее выполнения.

Сверхоперативная память — очень быстрое запоминающее устройство малого объема. Используется для компенсации разницы в скорости

обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Свойство — характеристика объекта, его параметр. Все объекты наделены определенными свойствами, которые в совокупности выделяют объект из множества других объектов.

Сервер — высокопроизводительный компьютер с большим объемом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования). См. также **Клиент**.

Сетевая операционная система — система, осуществляющая реализацию протоколов с реализацией управления серверами.

Сеть компьютерная — совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети. По степени географического распространения сети подразделяются на локальные, глобальные, городские и др. *Локальная сеть* связывает ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия. *Глобальная сеть* соединяет компьютеры, удаленные географически на большие расстояния друг от друга; отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.). *Городская сеть* обслуживает информационные потребности большого города.

Синтаксис — набор правил построения фраз языка, позволяющий определить, какие комбинации символов являются осмысленными предложениями в этом языке.

Система команд — совокупность операций, выполняемых некоторым компьютером.

Система — любой объект, который одновременно рассматривается и как единое целое, и как совокупность разнородных объектов, объединенных для достижения определенного результата.

Система программирования — система для разработки новых программ на конкретном языке программирования. Предоставляет пользователю мощные и удобные средства разработки программ: транслятор, редактор текстов программ, библиотеки стандартных программ, отладчик и др.

Система счисления — совокупность приемов и правил, по которым записываются и читаются числа.

Система телеконференций — основанная на использовании вычислительной техники система, позволяющая пользователям, несмотря на их взаимную удаленность в пространстве, а иногда и во времени, участвовать в совместных мероприятиях, таких как организация и управление сложными проектами.

Система управления базами данных (СУБД) — система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ конечных пользователей.

Системные программы. Программы общего пользования, выполняемые вместе с прикладными и служащие для управления ресурсами компьютера: центральным процессором, памятью, вводом-выводом.

Системы автоматизированного проектирования (САПР) — комплексные программно-технические системы, предназначенные для выполнения проектных работ с применением математических методов. Широко используются в архитектуре, электронике, механике и т.д. В качестве входной информации в САПР используются технические знания специалистов, которые вводят проектные требования, уточняют результаты, проверяют полученную конструкцию, изменяют ее и т.д. В САПР накапливается информация, поступающая из библиотек стандартов (данные о типовых элементах конструкций, их размерах, стоимости и др.).

Системы деловой графики — программные системы, позволяющие выводить на экран различные виды графиков и диаграмм.

Системы научной и инженерной графики — программные системы, позволяющие в цвете и в заданном масштабе отображать на экране графики двумерных и трехмерных функций, заданных в табличном или аналитическом виде; системы изолиний, в том числе и нанесенные на поверхность объекта, сечения, проекции, карты и др.

Сканер — устройство для ввода в компьютер документов: текстов, чертежей, графиков, рисунков, фотографий. Создает оцифрованное изображение документа и помещает его в память компьютера.

Событие — изменение состояния объекта.

Сопровождение программ — работы, связанные с обслуживанием программ в процессе их эксплуатации.

Сортировка — распределение упорядоченных данных по возрастанию или убыванию значений признака сортировки; упорядочение записей файла по одному или нескольким ключам.

Стример — устройство для резервного копирования больших объемов информации. В качестве носителя применяются кассеты с магнитной лентой емкостью 12 Гбайт и больше.

Структурное программирование — метод разработки программ, в частности, требующий разбиения программы на небольшие независимые части (модули). Обеспечивает возможность проведения строгого доказательства правильности программ, повышает уверенность в правильности конечной программы.

Сумматор — электронная логическая схема, выполняющая суммирование двоичных чисел.

Суперкомпьютер — очень мощный компьютер с производительностью свыше 100 мегафлоп (1 мегафлоп — миллион операций с плавающей точкой в секунду). Представляет собой многопроцессорный и (или) многомашинный комплекс, работающий на общую память и общее поле внешних устройств. Архитектура основана на идеях параллелизма и конвейеризации вычислений.

Схема алгоритма (блок-схема) — графическое представление алгоритма в виде последовательности блоков, соединенных стрелками.

Счетчик команд — регистр устройства управления, содержимое которого соответствует адресу очередной выполняемой команды. Служит для автоматической выборки команд программы из последовательных ячеек памяти.

Таблица истинности — табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений

истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

Текстовый редактор — программа для ввода и изменения текстовых данных (документов, книг, программ и т.д.). Обеспечивает редактирование строк текста, контекстный поиск и замену частей текста, автоматическую нумерацию страниц, обработку и нумерацию сносок, выравнивание краев абзаца, проверку правописания слов и подбор синонимов, построение оглавлений, распечатку текста на принтере и др.

Тест — некоторая совокупность данных для программы, а также точное описание всех результатов, которые должна выработать программа на этих данных в том виде, в котором эти результаты должны быть выданы программой.

Тестирование — этап решения задачи на компьютере, в процессе которого проверяется работоспособность программы, не содержащей явных ошибок.

Тип данных — понятие языка программирования, определяющее структуру констант, переменных и других элементов данных, а также разрешенные их значения и операции, которые можно над ними выполнять.

Топология компьютерной сети — логический и физический способ соединения компьютеров, кабелей и других компонентов, в целом составляющих сеть. Топология характеризует свойства сетей, не зависящие от их размеров. При этом не учитывается производительность и принцип работы этих объектов, их типы, длины каналов, хотя при проектировании эти факторы очень важны. Наиболее распространенные виды топологий: линейная, кольцевая, древовидная, звездообразная, ячеистая, полносвязная.

Топологии — раздел математики, изучающий свойства фигур, не изменяющиеся при любых деформациях, производимых без разрывов и склеиваний. Понятие топологии широко используется при создании компьютерных сетей.

Транслятор — программа-переводчик. Преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трафик — поток сообщений в сети передачи данных.

Трекбол — устройство управления курсором. Небольшая коробка с шариком, встроенным в верхнюю часть ее корпуса. Пользователь рукой вращает шарик и перемещает соответственно курсор.

Третье поколение компьютерной техники — семейства программно-совместимых машин с развитыми операционными системами. Обеспечивают мультипрограммирование. Быстродействие внутри семейства — от нескольких десятков тысяч до миллионов операций в секунду. Емкость оперативной памяти — несколько сотен тысяч слов. Элементная база — интегральные схемы.

Триггер — электронная схема, широко применяемая в регистрах компьютера для надежного запоминания одного бита информации. Имеет два устойчивых состояния, которые соответствуют двоичной единице и двоичному нулю.

Упаковщики (архиваторы) — программы, позволяющие записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл.

Устройство управления (УУ) — часть процессора, выполняющая функции управления устройствами компьютера. Файл. Именованная совокупность любых данных, размещенная на внешнем запоминающем устройстве и хранимая, пересылаемая и обрабатываемая как единое целое. Файл может содержать программу, числовые данные, текст, закодированное изображение и т.д. Имя файла регистрируется в каталоге.

Файл-сервер — содержит базу данных и программы управления данными для обеспечения многопользовательских запросов.

Цикл — прием в программировании, позволяющий многократно повторять одну и ту же последовательность команд (операторов).

Четвертое поколение компьютерной техники — поколение машин, разработанных после 1970 г. Эти компьютеры проектировались в расчете на эффективное использование современных высокоуровневых языков и упрощение процесса программирования для конечного пользователя. Элементная база — интегральные схемы. Емкость ОЗУ — десятки мегабайт. Машины этого поколения представляют собой персональные компьютеры либо многопроцессорные и (или) многомашинные комплексы, работающие на общую память и общее поле внешних устройств. Быстродействие — до нескольких сотен миллионов операций в секунду.

Шлюз — устройство соединения сети с другими сетями, например ЛВС с глобальной сетью.

Экспертная система — комплекс компьютерного программного обеспечения, помогающий человеку принимать обоснованные решения. Использует информацию, полученную заранее от экспертов — людей, которые в какой-либо области являются лучшими специалистами. Хранит знания об определенной предметной области. Обладает комплексом логических средств для выведения новых знаний, выявления закономерностей, обнаружения противоречий и др.

Электронная почта — система пересылки сообщений между пользователями вычислительных систем, в которой компьютер берет на себя все функции по хранению и пересылке сообщений.

Электронная таблица — программа, обрабатывающая таблицы, состоящие из строк и граф, на пересечении которых располагаются клетки. В клетках содержится числовая информация, формулы или текст. Значение в числовой клетке таблицы либо записано, либо рассчитано по формуле. В формуле могут присутствовать обращения к другим клеткам.

Электронный офис — интегрированный ППП, включающий в себя предметные программы и ИТ, обеспечивающие реализацию задач предметной области.

Язык ассемблера — система обозначений, используемая для представления в удобочитаемой форме программ, записанных в машинном коде. Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая называется ассемблером и является по сути простейшим транслятором.

Язык высокого уровня — язык программирования, более близкий к естественному языку, чем машинный код или язык ассемблера. Каждый

оператор в нем соответствует нескольким командам машинного кода или языка ассемблера.

ASCII — американский стандартный код обмена информацией. Широко используется для кодирования в виде байта букв, цифр, знаков операций и других компьютерных символов.

Host-ЭВМ — ЭВМ, установленная в узлах сети и решающая вопросы коммутации сети, доступа к сетевым ресурсам: модемам, факс-модемам, большим ЭВМ и др.; главная, ведущая, центральная ЭВМ.

HTML (HyperText Markup Language) — язык гипертекстовой разметки, используемый для создания документов, содержащих специальные команды форматирования и гипертекстовые ссылки и предназначенных для размещения в системе WWW.

HTTP (HyperText Transfer Protocol) — протокол передачи гипертекста, служащий для установления связи с документами HTML независимо от его местонахождения с помощью указателей URL.

Hub — концентратор, хаб. Устройство, обеспечивающее преобразование передаваемых сигналов таким образом, что сеть может быть дополнена новыми рабочими станциями.

OLE-технология (Object Linked and Embedding) — технология, позволяющая вставлять ранее созданные объекты (рисунки, графики, таблицы) во вновь создаваемые.

SILK (Speech Image Language Knowledge) — интерфейс, обеспечивающий перемещение на экране по речевой команде от одних поисковых образов к другим по смысловым (семантическим) связям.

SMTP (Simple Management Transfer Protocol) — упрощенный протокол передачи почтовых сообщений системами электронной почты.

SQL (Structured Query Language) — специализированный язык структурированных запросов, используемый при организации запросов, обновлении и управлении реляционными базами данных.

TCP/IP (Transmission Control Protocol / Internet Protocol) — протокол управления передачей сообщений — межсетевой протокол, охватывающий средства доступа к среде, средства транспортировки пакетов, сеансовые коммуникационные средства, средства передачи файлов, средства электронной почты и средства терминальной эмуляции.

URL (Uniform Resource Locator) — универсальный локатор ресурсов, характеризующий полный путь к конкретному документу или разделу на компьютере, подключенном к сети Интернет, а также метод доступа к нему, т.е. протокол работы с программами-серверами, функционирующими на удаленном компьютере.

Web page — Web-страница в системе WWW.

WIMP (Windows Image Menu Pointer) — интерфейс, обеспечивающий размещение на экране окон, содержащих образы программ и меню действий.

WWW (World Wide Web) — глобальная сетевая система, объединяющая серверы Internet, «всемирная паутина».

СПИСОК ПОЛЕЗНЫХ ИНТЕРНЕТ-ССЫЛОК

1. <http://www.informika.ru/> — Сайт Государственного научно-исследовательского института информационных технологий и телекоммуникаций.
2. <http://www.citforum.ru/> — Центр информационных технологий.
3. <http://www.5ballov.ru/> — Образовательный портал.
4. <http://www.fio.ru/> — Федерация Интернет-образования.
5. <http://tests.academy.ru/> — Тесты из области информационных технологий.
6. <http://www.codenet.ru/> — Все для программиста.
7. <http://public.tsu.ru/~wawlasov/start.htm> — В помощь учителю информатики.
8. <http://sciedu.city.ru/> — Наука и образование в России.
9. <http://www.ed.gov.ru/> — Сайт Министерства образования Российской Федерации.
10. <http://iit.metodist.ru/> — Лаборатория информационных технологий.
11. <http://schools.keldysh.ru/sch444/MUSEUM/> — Виртуальный музей информатики.
12. <http://www.otd.tstu.ru/direct1/inph.html> — Сайт, посвященный информатике.
13. <http://www.inr.ac.ru/~info21/> — ИНФОРМАТИКА-21. Международный научно-образовательный проект от Российской Академии Наук.
14. <http://www.morepc.ru/> — Информационно-справочный портал.
15. <http://www.ito.su/> — Информационные технологии в образовании.
16. <http://www.inftech.webservis.ru/> — Статьи по информационным технологиям.

СПИСОК ЛИТЕРАТУРЫ

1. Алексеев В. Е., Ваулин А. С., Петрова Г. Б. Вычислительная техника и программирование. Практикум по программированию. — М.: Высш. шк., 1991.
2. Безручко В. Т. Практикум по курсу «Информатика». — М.: Финансы и статистика, 2002.
3. Бордовский Г. А. Информатика в понятиях и терминах. — М.: Просвещение, 1991.
4. Борисов А. В. Самоучитель по работе с компьютерной сетью. Пособие для начинающих и опытных пользователей ПК. — М.: Альянс-пресс, 2003.
5. Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989.
6. Власов В. К. Элементы информатики. — М.: Наука, 1988.
7. Волченков Н. Г. Программирование на Visual Basic 6. — М.: ИНФРА-М, 2002.
8. Вычислительная техника и программирование / Под ред. А. В. Ретрова. — М.: Высш. шк., 1990.
9. Глушаков С. В., Сурядный А. С. Программирование на Visual Basic 6.0. — Харьков: ФОЛНО, 2002.
10. Евдокимов В. В. Экономическая информатика. — СПб.: Питер Publishing, 1997.
11. Есеев Г. А., Пацук С. Н., Симонович С. В. Вы купили компьютер. Полное руководство для начинающих в вопросах и ответах. — М.: АСТ-Пресс: ИнфоркомПресс, 1998.
12. Жаров А. Желесо IBM. — М.: МИКРОАРТ, 1996.
13. Завгородний В. И. Комплексная защита информации в компьютерных системах. — М.: Логос, 2001.
14. Интернет. Самоучитель / А. Денисов, В. Вихарев, А. Белев, Г. Наумов. — М.: Питер, 2003.
15. Информатика. 10—11 класс / Под ред. Н. В. Макаровой. — СПб.: Питер, 1999.
16. Информатика. Задачник-практикум / Под ред. И. Г. Семакина, Е. К. Хеннера. — М.: Лаборатория Базовых Знаний, 1999.
17. Информатика. Терминологический словарь. — М.: Всероссийский НИИ комплексной информации по стандартизации и качеству, 1992.
18. Информатика / Под ред. Н. В. Макаровой. — М.: Финансы и статистика, 2002.
19. Информатика / Под ред. П. П. Бельского. — Ростов-на-Дону: Феникс, 2002.
20. Калининский Н. А., Галевский Г. В. Эффективные технологии работы в сети Интернет. — М.: Флинта: Наука, 2004.

21. Каранчук В. П. Основы применения ЭВМ. — М.: Радио и связь, 1988.
22. Кузнецов А. А. Основы информатики. — М.: Дрофа, 1998.
23. Кушниренко А. Г. Информатика. — М.: Дрофа, 1998.
24. Левин В. И. Носители информации в цифровом веке. — М.: КомпьютерПресс, 2000.
25. Леонтьев В. П. Новейшая энциклопедия персонального компьютера. — М.: ОЛМА-ПРЕСС Образование, 2004.
26. Ляхович В. Ф., Крамаров С. О. Основы информатики: Учеб. пособие для студ. средних специальных учебных заведений. — М.: Просвещение, 1997.
27. Хэлворсон М., Янг М. Эффективная работа с Office XP. — СПб.: Питер, 2003.
28. Назаров С. В., Мельников П. П. Программирование на MS Visual Basic. — М.: Финансы и статистика, 2001.
29. Острейковский В. А. Информатика. — М.: Высш. шк., 1999.
30. Першиков В. И., Савинов В. М. Толковый словарь по информатике. — М.: Финансы и статистика, 1991.
31. Семакин И. Г., Хеннер Е. К. Информатика. 11-й класс. — М.: Бином: Лаборатория Базовых Знаний, 2002.
32. Толковый словарь по вычислительным системам / Под ред. В. Иллингвотера. — М.: Машиностроение, 1990.
33. Угринович Н. Д. Информатика и информационные технологии. Практикум для 10—11 классов. — М.: Бином: Лаборатория Базовых Знаний, 2003.
34. Угринович Н. Д. Информатика и информационные технологии: Учебник для 10—11 классов. — М.: Бином: Лаборатория Базовых Знаний, 2003.
35. Фигурнов В. Э. IBM PC для пользователя. Краткий курс. — М.: ИНФРА-М, 2003.
36. Фролов Г. Д., Кузнецов Э. И. Элементы информатики. — М.: Высш. шк., 1989.
37. Шауцкова Л. Э. Информатика: Учеб. пособие для 10—11 классов. — М.: Просвещение, 2002.
38. Язык компьютера: Пер. с англ. — М.: Мир, 1989.
39. Федеральный закон «Об информатизации и защите информации» от 20.02.1995 № 24-03.
40. ГОСТ 34.003—90 Информационная технология. Комплекс стандартов на «автоматизированные системы».

ОГЛАВЛЕНИЕ

Предисловие	3
Глава 1. Информация и информационные процессы	7
1.1. Введение в информатику	7
1.2. Информация в реальном мире	10
1.3. Информационные системы	14
1.4. Информация и ее свойства	20
1.4.1. Измерение информации	21
1.4.2. Кодирование информации	28
Глава 2. Общие принципы организации и работы персонального компьютера	30
2.1. Представление информации в персональном компьютере	30
2.1.1. Системы счисления	30
2.1.2. Перевод чисел из системы счисления в десятичную с основанием q	32
2.1.3. Перевод чисел из десятичной системы счисления в систему счисления с основанием q	33
2.1.4. Системы счисления, используемые в компьютере	35
2.1.5. Арифметические операции в позиционных системах счисления	36
2.1.6. Представление чисел в компьютере	40
2.1.7. Представление других видов информации в компьютере	43
2.2. Логические основы построения компьютера	49
2.2.1. Алгебра логики	50
2.2.2. Основные законы алгебры логики	54
2.2.3. Логические основы устройства компьютера	58
2.2.4. Триггер	61
2.2.5. Сумматор двоичных чисел	62
2.3. Программное управление компьютером	64
2.3.1. Принципы фон Неймана	65
2.3.2. Структура и виды команд	67
2.3.3. Архитектура компьютера	69
Глава 3. Компьютер и программное обеспечение	75
3.1. Аппаратная реализация компьютера	75
3.1.1. Процессор (микропроцессор)	78
3.1.2. Оперативная память	79
3.1.3. Контроллеры и системная магистраль	81

3.1.4. Хранение информации. Внешняя память	82
3.1.5. Устройства ввода информации	89
3.1.6. Устройства вывода информации	92
3.1.7. Устройства ввода-вывода информации	97
3.1.8. Компьютеры четвертого и пятого поколения	99
3.1.9. Безопасная работа на компьютере	100
3.2. Компьютерные сети	102
3.2.1. Классификация компьютерных сетей	104
3.2.2. Топология сети	105
3.2.3. Способы коммутации и передачи данных	106
3.2.4. Характеристики коммуникационной сети	108
3.2.5. Архитектура компьютерной сети	109
3.2.6. Аппаратные средства организаций компьютерной сети	110
3.2.7. Глобальная сеть Интернет	114
3.2.8. Обеспечение защиты информации в компьютерных сетях	116
3.3. Программное обеспечение персонального компьютера	118
3.3.1. Системное программное обеспечение	119
3.3.2. Пакеты прикладных программ (ППП)	121
3.3.3. Системы программирования	122
3.3.4. Операционная система	123
3.3.5. Логическая структура диска	128
3.3.6. Загрузка операционной системы	134
3.3.7. Основные свойства и возможности Windows	136
Глава 4. Системное программное обеспечение	140
4.1. Работа в операционной системе Windows	140
4.1.1. Управление Windows с помощью мыши	141
4.1.2. Элементы интерфейса Windows	142
4.2. Программы обслуживания магнитных дисков	159
4.2.1. Форматирование дисков	160
4.2.2. Копирование гибких дисков	161
4.2.3. Определение объема свободного пространства диска	162
4.2.4. Проверка рабочих поверхностей дисков	162
4.2.5. Дефрагментация диска	164
4.2.6. Очистка диска	165
4.3. Программы-архиваторы	166
4.4. Компьютерные вирусы и антивирусные программы	171
Глава 5. Информационные технологии	178
5.1. Информационные технологии	178
5.2. Технология обработки текстовой информации	180
5.2.1. Ввод текста	187
5.2.2. Редактирование текста	187
5.2.3. Форматирование текста	190
5.2.4. Работа с иллюстрациями	194
5.2.5. Работа с таблицами	196
5.2.6. Колонтитулы	198
5.2.7. Сохранение документа	198

5.2.8. Печать документа	199
5.3. Технология обработки графической информации	203
5.3.1. Растровая графика	203
5.3.2. Векторная графика	205
5.3.3. Форматы графических файлов	206
5.3.4. Графические редакторы	207
5.3.5. Примеры создания графического изображения	211
5.4. Технология обработки числовой информации	213
5.4.1. Типы данных, используемых в Microsoft Excel	216
5.4.2. Относительные и абсолютные ссылки	217
5.4.3. Табличные вычисления	218
5.4.4. Встроенные функции	221
5.4.5. Сортировка и поиск данных	223
5.4.6. Построение диаграмм и графиков функций	226
5.5. Технология хранения, поиска и сортировки информации	233
5.5.1. Типы баз данных	233
5.5.2. Система управления базами данных	237
5.6. Компьютерные коммуникации	262
5.6.1. Работа с электронной почтой	263
5.6.2. Всемирная информационная сеть Интернет	270
5.6.3. Поисковые системы	276
5.6.4. Общение в реальном времени	281
5.6.5. Работа с файловыми архивами	283
5.7. Мультимедийные технологии	286
Глава 6. Моделирование и формализация	293
6.1. Материальные и информационные модели	294
6.1.1. Модель	294
6.1.2. Формализация	296
6.1.3. Визуализация формальных моделей	299
6.1.4. Модель как совокупность объектов системы	300
6.1.5. Типы информационных моделей	301
6.2. Информационная технология решения задач	304
Глава 7. Основы алгоритмизации	310
7.1. Алгоритм и его свойства	310
7.2. Базовые алгоритмические конструкции	315
7.2.1. Алгоритмы линейной структуры	316
7.2.2. Алгоритмы разветвляющейся структуры	317
7.2.3. Алгоритмы циклической структуры	322
7.2.4. Массивы	327
7.2.5. Вспомогательные алгоритмы	330
Глава 8. Программирование на Visual Basic 6	333
8.1. Языки программирования	333
8.2. Концепция программирования на Visual Basic	337
8.2.1. Структура объектов	337
8.2.2. Объектно-ориентированный язык программирования Visual Basic	339

8.3. Выражения и функции	348
8.3.1. Объявление переменных и присваивание им значений	350
8.3.2. Выражения	354
8.3.3. Функции в языке Visual Basic	355
8.4. Программирование ветвлений и повторений	364
8.4.1. Программирование ветвлений	364
8.4.2. Программирование повторений	369
8.5. Массивы	375
8.6. Графические методы	381
8.6.1. Графические методы Scale и PSet	381
8.6.2. Установка цвета	385
8.6.3. Использование графического метода Line и метода Print	387
8.6.4. Метод Circle	390
8.6.5. Работа с изображениями	392
Приложение 1. Глоссарий	394
Приложение 2. Список полезных Интернет-ссылок	408
Список литературы	409

Учебное издание

**Колмыкова Елена Алексеевна,
Кумскова Ирина Александровна**

Информатика

Учебное пособие

10-е издание, стереотипное

Редактор *И. В. Мочалова*
Технический редактор *О. Н. Крайнова*
Компьютерная верстка: *Л. М. Белова*
Корректоры *Т. В. Кузьмина, И. В. Мошелева*

Изд. № 110107936. Подписано в печать 30.07.2012. Формат 60×90/16.
Гарнитура «Таймс». Печать офсетная. Вумга офсетная № 1. Усл. печ. л. 26,0.
Тираж 2000 экз. Заказ № 1103.

ООО «Издательский центр «Академия», www.akademika-publishing.ru
125252, Москва, ул. Зорге, д. 15, корп. 1, пом. 366.
Адрес для корреспонденции: 129083, Москва, пр-т Мира, 101В, стр. 1, а/я 48.
Тел./факс: (495) 648-0307, 616-00-29.
Санитарно-эпидемиологическое заключение № РОСС RU. АЛ51. Н 16067 от 06.03.2012.

Отпечатано с электронных носителей издательства.
ОАО «Тверской полиграфический комбинат», 170024, г. Тверь, пр-т Ленина, 8.
Телефон: (4822) 44-52-03, 44-50-34. Телефон/факс: (4822) 44-43-18.
Home page — www.tverpk.ru Электронная почта (E-mail) — info@tverpk.ru

ИНФОРМАТИКА



Издательский центр
«Академия»
www.academia-moscow.ru