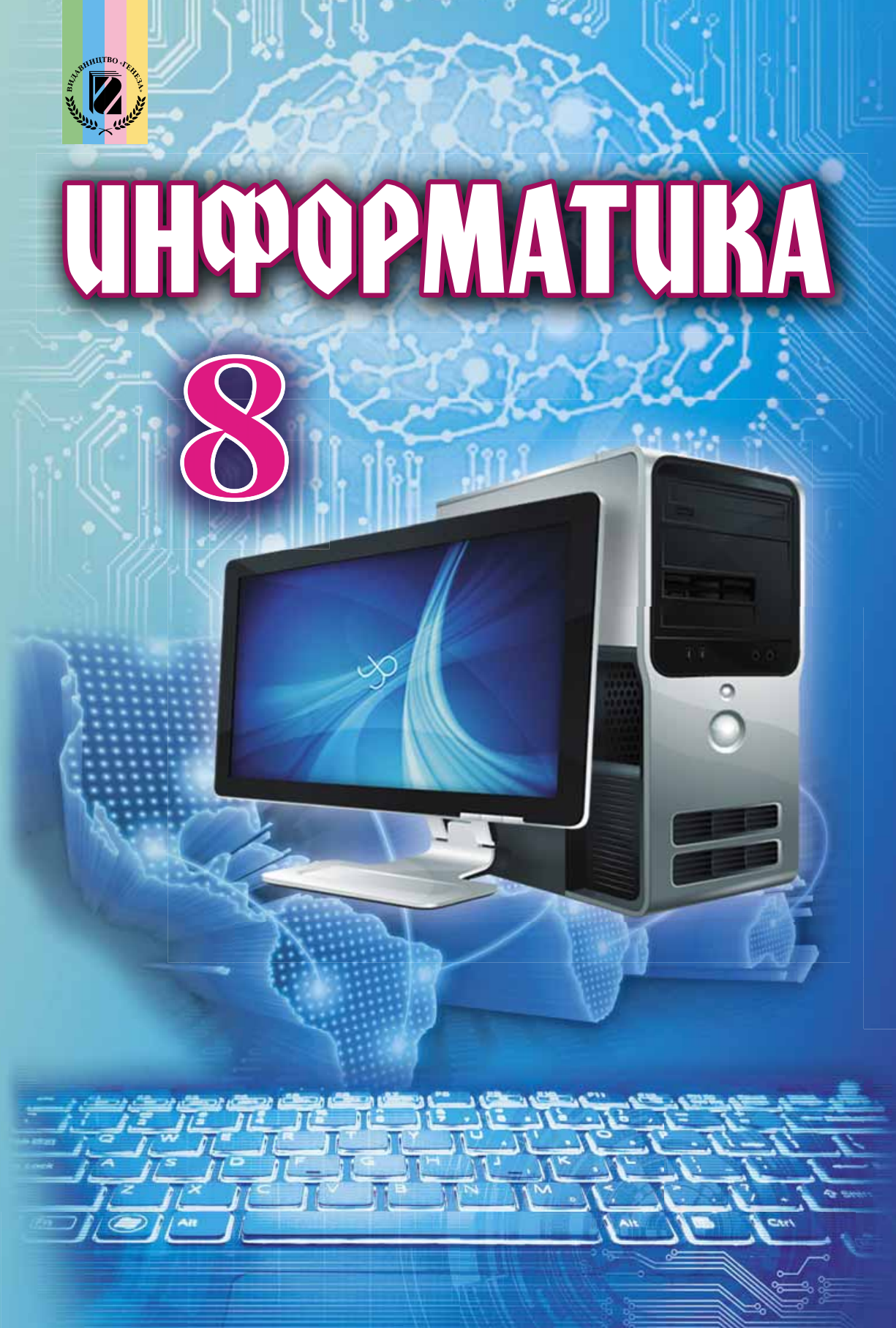
ВИДАВНИЦТВО «ГЕНЕЗА»

ІНФОРМАТИКА

8



УДК 004(075.3)
ББК 32.97я721
И74

Авторы:

И. Я. РИВКИНД, Т. И. ЛЫСЕНКО, Л. А. ЧЕРНИКОВА, В. В. ШАКОТЬКО

Рекомендовано Министерством образования и науки Украины
(Приказ МОН Украины от 10.05.2016 № 491)

Издано за счет государственных средств. Продажа запрещена

Переведено по изданию:

Інформатика: підруч. для 8-го кл. загальноосвіт. навч. закл. / Й. Я. Ривкінд [та ін.]. – Київ: Генеза, 2016. – 288 с. : іл. – ISBN 978-966-11-0692-4.

Эксперты, осуществившие экспертизу данного учебника во время проведения конкурсного отбора проектов учебников для учащихся 8 класса общеобразовательных учебных заведений и сделавшие вывод о целесообразности предоставления учебнику грифа «Рекомендовано Министерством образования и науки Украины»:

Жук Н.А., методист Гощанского районного методического кабинета Ривненской области;

Попова Л.Н., учитель информатики специализированной общеобразовательной школы I–III ступеней № 7 Светловодского городского совета Кировоградской области.

Информатика : учебн. для 8-го кл. общеобразоват.
И74 учебн. завед. / И.Я. Ривкинд [и др.]. — Киев : Генеза, 2016. — 288 с. : илл.
ISBN 978-966-11-0754-9.

Материал учебника, согласно новой программе, разделен на 8 глав. Главы «Текстовый процессор» и «Табличный процессор» представлены для пакета **Microsoft Office**.

Каждый пункт учебника содержит вопросы для актуализации знаний, основной учебный материал, тренировочные упражнения, вопросы для самоконтроля и практические задания, распределенные в соответствии с уровнями учебных достижений учащихся. Учебник содержит 16 практических работ.

Для повышения интереса к изучению предмета, кроме основного материала, учебник содержит рубрики: «Для тех, кто хочет знать больше», «Знаете ли вы, что...».

УДК 004(075.3)
ББК 32.97я721

ISBN 978-966-11-0754-9 (рус.)
ISBN 978-966-11-0692-4 (укр.)

© Ривкинд И.Я., Лысенко Т.И., Черникова Л.А., Шакотько В.В., 2016
© Издательство «Генеза», оригинал-макет, 2016

ДОРОГИЕ ДРУЗЬЯ!

Вы продолжаете изучение интересного и важного предмета – **информатики**.

В предыдущих классах вы уже научились выполнять операции с объектами операционной системы компьютера, создавать информационные модели, обрабатывать графические, текстовые, числовые и мультимедийные данные, искать сведения и переписываться в Интернете, составлять алгоритмы для исполнителя **Рыжий кот**, решать компетентностные задачи и выполнять учебные проекты.

На уроках информатики в 8-м классе вы ознакомитесь с кодированием данных, рассмотрите аппаратное и программное обеспечение компьютера, продолжите обрабатывать текстовые и числовые данные, решать компетентностные задачи. Вы научитесь создавать видеофайлы, обрабатывать аудиоданные, разрабатывать и выполнять программные проекты в среде программирования. Умение применять приобретенные знания вы отработаете, выполняя учебные проекты. И в этом вам поможет наш учебник.

Материал учебника разделен на главы. Каждая глава состоит из пунктов, которые, в свою очередь, содержат подпункты. В начале каждого пункта приведены вопросы для повторения изученного. Ответы на них облегчат понимание и усвоение нового материала. Эти вопросы обозначены  .

Внимательно читайте материал, изложенный в учебнике. Важный материал собран в конце каждого пункта в отдельной рубрике  **«Самое важное в этом пункте»**.

В конце каждого пункта размещены вопросы для самоконтроля  **«Дайте ответы на вопросы»**. Рекомендуем вам дать на них ответы после изучения учебного материала пункта. Возле каждого вопроса стоит значок, указывающий на то, что правильный ответ на этот вопрос соответствует:

° – начальному и среднему уровням учебных достижений;

• – достаточному уровню учебных достижений;

* – высокому уровню учебных достижений.

Так же обозначены и уровни практических заданий в рубрике  **«Выполните задание»**, после каждого пункта. Задания, рекомендованные авторами для работы дома, обозначены  . Над заданиями с пометкой  целесообразно поработать в парах или небольших группах.

Кроме основного материала, пункты учебника содержат рубрики:

-  «Для тех, кто хочет знать больше»;
-  «Знаете ли вы, что...»;
-  «Интересные факты из истории»;
-  «Для тех, кто работает с **ОС Linux**»;
-  «Для тех, кто работает с **LibreOffice**»;
-  «Для тех, кто работает с **Microsoft Office 2010**»;
- «Толковый словарь терминов и понятий» (на зеленом фоне).

В конце учебника есть **Словарик**.

Файлы-заготовки для выполнения заданий и практических работ размещены в Интернете по адресу *http://allinf.at.ua*

Желаем вам успехов в изучении самой интересной и современной науки – ИНФОРМАТИКИ!

Авторы

Глава 1. Кодирование данных

В этой главе вы узнаете:

- об обработке данных как информационном процессе
- о кодировании и декодировании сообщений
- о таблицах кодов символов
- о двоичном кодировании
- о единицах измерения длины двоичного кода

1.1. ОБРАБОТКА ДАННЫХ КАК ИНФОРМАЦИОННЫЙ ПРОЦЕСС. КОДИРОВАНИЕ И ДЕКОДИРОВАНИЕ СООБЩЕНИЙ

1. Какие информационные процессы вы знаете? В чем сущность процесса обработки сообщений?
2. Какие существуют способы представления сообщений?
3. Что такое данные? Приведите примеры данных.

ОБРАБОТКА ДАННЫХ КАК ИНФОРМАЦИОННЫЙ ПРОЦЕСС

В современном обществе немало профессий связано с обработкой данных, представленных словами, числами, графическими изображениями и т. п. Бухгалтеры начисляют заработную плату, исходя из данных об объемах выполненных работ. Врачи ставят диагноз пациентам на основе результатов различных анализов. Синоптики составляют прогноз погоды, анализируя данные о движении воздушных масс. Астрономы рассчитывают траектории движения космических объектов по данным об их расположении. Художник создает иллюстрации к книге, ознакомившись с ее содержанием (рис. 1.1). Выполняют обработку данных и ученики, решая задачи, готовя сочинения, строя графики функций и т. п.



Рис. 1.1. Обработка данных представителями разных профессий

Как вы уже знаете, **обработка данных** – это процесс получения новых данных из имеющихся.

Так, если мы решаем математическую задачу, то из имеющихся (начальных, входных) данных условия задачи мы получаем новые (выходные) данные – решение задачи. При определении победителей легкоатлетического забега обработка данных состоит в размещении фамилий участников забега в списке в соответствии со временем преодоления ими дистанции от меньшего до большего значения времени. При построении диаграммы в табличном процессоре из имеющихся числовых данных получаются новые данные, представленные графическим способом.

Обрабатывать данные можно с использованием различных устройств, в том числе с использованием компьютера. Для этого нужно подать данные в виде, пригодном для обработки устройствами компьютера, и определить правила, по которым должна происходить обработка.

КОДИРОВАНИЕ СООБЩЕНИЙ

С целью сохранения, передачи, обработки, защиты сообщений часто меняют способ их представления.

Устные сообщения записывают на бумаге, заменяя звуки человеческой речи буквами алфавита. При этом выполняется сохранение сообщений. Во время разговора по телефону звуки преобразуются в электрические сигналы. Это делает возможной передачу сообщений на большие расстояния. Решая задачу на уроке математики, слова, обозначающие числа, записывают цифрами, а математические операции – специальными знаками. Это упрощает выполнение операций по обработке сообщений. Для защиты содержания сообщений от посторонних лиц проводят шифрование, заменяя буквы в тексте сообщения другими буквами, числами или условными обозначениями.

Процесс замены одной последовательности сигналов, которой представлено сообщение, другой последовательностью сигналов называют **кодированием сообщения**.

Код (лат. *codex* – книга) – система условных знаков, сигналов для передачи, обработки и хранения сообщений.

Кодирование широко используется в современной жизни для представления сообщений в виде, пригодном для обработки не только человеком, но и техническими устройствами. Так, сведения о товарах кодируют с использованием штрих-кодов (рис. 1.2). Цифрами на штрих-коде обозначены сведения о товаре, стране и предприятии – производителе этого товара. Этот числовой код предназначен для обработки человеком и используется для упорядоченного хранения данных о товарах. А толщина линий штрих-кода и промежутков между ними кодирует те же сведения. Этот графический код воспринимается специальными сканерами и обрабатывается дальше с использованием компьютера.



Рис. 1.2. Штрих-код товара

Для кодирования сообщений определяют не только набор символов, знаков, сигналов, которые будут использованы для замены сигналов заданного сообщения, но и правила, по которым происходит эта замена.

Например, из истории хорошо известен шифр, которым пользовался для секретной переписки со своими генералами римский император Гай Юлий Цезарь (100–44 гг. до н. э.). Правила кодирования состоят в том, что каждая буква в тексте сообщения заменяется другой, которая находится в алфавите на расстоянии нескольких позиций от заданной буквы (рис. 1.3).

Так, закодировав сообщение «*привет*» с использованием шифра Цезаря со сдвигом на 4 буквы вправо, получим сообщение «*уфмжйц*».

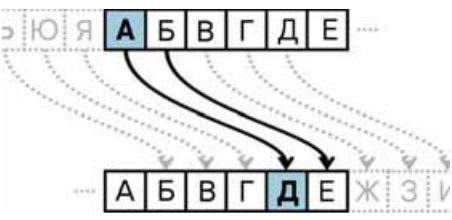


Рис. 1.3. Правила замены символов в шифре Цезаря со сдвигом на 4 буквы вправо

ДЕКОДИРОВАНИЕ СООБЩЕНИЙ

Процесс получения исходного сообщения из закодированного называют **декодированием** сообщения.

Декодирование сообщений происходит, когда мы читаем вслух напечатанный текст, исполняем музыкальное произведение по нотам, расшифровываем сообщения, определяем по штрих-коду страну, в которой изготовлен товар, и т. п.

Для декодирования сообщения, закодированного шифром Цезаря со сдвигом на 4 буквы вправо, нужно каждую букву закодированного сообщения заменить другой, расположенной в алфавите на 4 позиции слева от заданной. Таким образом, декодировав сообщение «*итефян ийса*», получим «*добрый день*».

Кодирование и декодирование сообщений – это примеры процессов обработки данных.

Самое важное в этом пункте

Обработка данных – это процесс получения новых данных из имеющихся.

Кодирование сообщения – это процесс замены одной последовательности сигналов, которой подано сообщение, другой последовательностью сигналов.

Кодирование сообщений выполняется с целью их хранения, передачи, обработки, защиты.

Для кодирования сообщений определяют набор сигналов, которые будут использованы для замены сигналов заданного сообщения, и правила, по которым происходит эта замена.

Декодирование сообщения – это процесс получения исходного сообщения из закодированного.

Кодирование и декодирование сообщений – это примеры процессов обработки данных.

Ответьте на вопросы

1°. В чем заключается процесс обработки данных?

2•. Какие новые данные могут быть получены в результате обработки данных:

а) о количестве и цене проданных в магазине товаров;

б) о росте учеников класса;

в) о расстоянии между двумя населенными пунктами и скорости движения поезда, курсирующего между ними?

3•. В чем заключается процесс кодирования сообщений?

4•. С какой целью кодируют сообщения?

5*. С какой целью звуки музыки кодируют нотами; сообщения об особенностях движения по дороге кодируют дорожными знаками?

6*. Сообщение о чем можно закодировать с использованием цветов; графических обозначений?

7•. Что следует предварительно определить для кодирования сообщения?

8*. По каким правилам в игре «Морской бой» кодируют положение кораблей? С какой целью?

9°. В чем заключается процесс декодирования сообщений?

Выполните задания

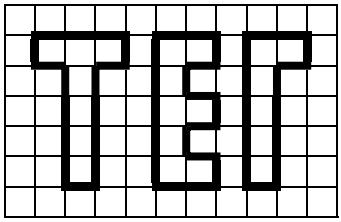
1•. Приведите примеры обработки данных.

2•. Приведите примеры кодирования сообщений, используемых в музыке, физике, химии, географии и др.

3•. Закодируйте с использованием шифра Цезаря со сдвигом на 4 буквы вправо сообщения:
а) *информатика;*
б) *моя школа;*
в) *кодирование и декодирование сообщений.*

4•. Сообщения закодированы шифром Цезаря со сдвигом на 3 буквы вправо. Декодируйте сообщения:
а) *фирхвдуя*;
б) *шсусыгв тсжсзг*;
в) *цъидрюм жсз ргьгофв.*

5•. Изображение, созданное карандашом на листе бумаги в клеточку, закодировано по следующему правилу: стрелка указывает направление, в котором перемещается карандаш по линии сетки, а число указывает количество клеточек для перемещения. Воссоздайте закодированное изображение:



а) ↑5 →3 ↓5 ←1 ↑4 ←1 ↓4 ←1;
б) ←3 ↓5 →3 ↑1 ←2 ↑1 →2 ↑1 ←2 ↑1 →2 ↑1.

6•. Закодируйте данные изображения по следующему правилу: стрелка указывает направление, в котором перемещается карандаш по линии сетки, а число – количество клеточек для перемещения.

7*. Дополните правила кодирования, приведенные в предыдущем задании, так, чтобы можно было обозначить перемещения карандаша, при которых не происходит рисования.

	1	2	3	4	5	6
1	А	Б	В	Г	Д	Е/Ё
2	Ж	З	И/Й	К	Л	М
3	Н	О	П	Р	С	Т
4	У	Ф	Х	Ц	Ч	Ш
5	Щ	Ы	Ь/Ъ	Э	Ю	Я

Квадрат Полибия

8•. Один из древнейших шифров разработал Полибий (III в. до н. э.) – греческий историк, полководец, государственный деятель. В шифре, который назвали «квадрат Полибия», каждая буква алфавита (или пара букв, соответствующих близким по произношению звукам) расположена в таблице. При кодировании сообщения каждая буква заменяется парой цифр – номерами столбца и строки таб-

лицы, на пересечении которых она расположена. Для кодирования сообщений на русском языке может быть использована таблица, изображенная на с. 9.

Декодируйте сообщения, закодированные по рассмотренным правилам с использованием приведенной таблицы:

а) 42 31 11 51 43 11 63; б) 32 13 24 23 43 62 11 63 32 42 11.

9•. Закодируйте сообщения, используя правила кодирования, приведенные в предыдущем задании:

а) *информатика*;

б) *сообщение*.

10*. Предложите свои правила кодирования символов текста и закодируйте по этим правилам сообщения:

а) *сегодня хороший день*;

б) *мне нравится футбол*.

11*. Найдите сведения и подготовьте сообщение о способах кодирования:

а) азбука Морзе; в) шрифт Брайля;

б) семафорная азбука; г) язык цветов «селам».

1.2. КОДИРОВАНИЕ СИМВОЛОВ

1. Что такое кодирование сообщений? С какой целью кодируют сообщения?
2. Приведите примеры кодирования сообщений.
3. Что такое декодирование сообщений?

КОДИРОВАНИЕ СИМВОЛОВ ТЕКСТА

Для обработки текстовых сообщений с использованием компьютера символы сообщения кодируют числами.

Телетайп (лат. *tele* – удаленный, англ. *type* – печать) – электромеханическая печатная машина, которую используют для передачи текстовых сообщений по проводам.

Для кодирования текста используют **таблицы кодов символов.** В них каждому символу, который может быть использован в текстовом сообщении, соответствует некоторое число. В 1963 г. в США был разработан набор таких кодов символов для передачи сообщений по телетайпу. Позже он стал стандартом для использования в компьютерной технике и получил название **таблицы кодов символов ASCII** (англ. *American Standard Code for Information Interchange* – американский стандартный код для обмена информацией).

В таблице **ASCII** буквам английского алфавита, цифрам, знакам препинания, символам редактирования и форматирования текста соответствуют числа от 0 до 127 (табл. 1.1).

Текстовые символы, которые кодируются в таблице **ASCII** числами от 32 до 127, приведены в приложении 1. Его можно найти на сайте **Інформатика для всіх** (*allinf.at.ua*) в разделе **Матеріали до підручників. 8 клас. Підручник.**

Таблица 1.1

Группы кодов символов таблицы ASCII

Диапазон кодов	*Группа символов*	*Пример кода*	*Соответствующий символ*
От 0 до 31, 127	Специальные символы	10	Символ, соответствующий переносу курсора на новую строку
		13	Символ, соответствующий возврату курсора в начало строки
		27	Символ, соответствующий нажатию клавиши **Esc**
От 32 до 64, от 91 до 96, от 123 до 126	Знаки препинания и цифры	32	пробел
		48	цифра *0*
		123	{
От 65 до 90	Большие буквы английского алфавита	65	*A*
		66	*B*
		90	*Z*
От 97 до 122	Малые буквы английского алфавита	97	*a*
		98	*b*
		122	*z*

Таблица кодов символов **ASCII** содержит коды букв только английского алфавита. Для кодирования букв других алфавитов были разработаны другие таблицы кодов символов. Например, таблицы **КОИ-8 (КОИ** – **к**од **о**бмена **и**нформацией) и **Windows-1251** содержат без изменений все коды таблицы **ASCII**, а также – коды букв кириллицы. Этим буквам соответствуют натуральные числа от 128 до 255.

Кириллица – алфавит для группы славянских языков, к которой относятся украинский, русский, белорусский, болгарский и другие языки.

Английский и некоторые другие западноевропейские языки используют алфавит, который называют **латиницей**.

Таблица кодов символов **Windows-1251** является стандартом для кодирования букв кириллицы в операционной системе **Windows**. В ней, например, букве «*а*» русского алфавита ставится в соответствие число 224, букве «*б*» – число 225, букве «*я*» – число 255 и др. Текстовые символы, которые кодируются в таблице **Windows-1251** числами от 128 до 255, приведены в приложении 2. Его можно найти на сайте **Інформатика для всіх** (*allinf.at.ua*) в разделе **Матеріали до підручників. 8 клас. Підручник.**

Целых чисел от 0 до 255 достаточно, чтобы закодировать символы двух алфавитов – латиницы и кириллицы, а также некоторые другие символы. Но для кодирования символов других алфавитов (греческого или арабского, иероглифов и т. п.) требуется значительно больше значений кодов. Для них разработана таблица кодов символов **Юникод** (англ. *Unicode* – **уни**фицированное **код**ирование).

Таблица **Юникод** состоит из 17 наборов по 65 536 значений кодов в каждом и дает возможность закодировать 1 114 112 разных символов, то есть почти все символы письменности всех мировых языков. Как и в других таблицах кодов, в **Юникоде** неизменными остаются первые 128 значений кодов, соответствующих таблице **ASCII**. Отдельный раздел в таблице **Юникод** содержит коды букв кириллицы. Например, букве «*а*» русского алфавита соответствует код 53 424, букве «*б*» – код 53 425, букве «*я*» – код 53 455 и др.

Сейчас в новейших операционных системах используется таблица кодов **Юникод.**

Сравнение структур различных таблиц кодов символов приведено в таблице 1.2.

Таблица 1.2

Сравнение структуры таблиц кодов символов

Таблица кодов символов	*Количество кодов символов*	*Символы с кодами от 0 до 127*	*Символы с кодами от 128 до 255*	*Символы с кодами, большими, чем 255*
ASCII	128	Буквы английского алфавита, цифры, знаки препинания, специальные символы	Нет	Нет
КОИ-8	256	Как в таблице **ASCII**	Символы кириллицы и некоторые другие символы	Нет
Windows-1251	256	Как в таблице **ASCII**	Символы кириллицы и некоторые другие символы	Нет
Юникод	1 114 112	Как в таблице **ASCII**	Символы алфавитов разных языков мира и некоторые другие символы	

Для тех, кто хочет знать больше

В таблице **Юникод** содержатся коды не только букв и цифр, но и символов, которые обозначают торговые марки, денежные единицы, символы транскрипций, идеограммы и т. п. Например, кодом символа украинской денежной единицы гривни ₴ является число 8 372, кодом идеограммы *мужчина* 🚹 является число 10 080, а кодом идеограммы *женщина* 🚺 – число 10 081 и т. п.

Для вставки символов в документ можно применять их коды из таблицы **Юникод**. Для этого следует нажать и удерживать клавишу **Alt** и набрать код символа

на дополнительной цифровой клавиатуре. Так могут быть вставлены, например, символы *снеговик* ☃ – код 9731, *сердце в виде цветка* ❧ – код 10087, *нота* ♪ – код 9834, *шахматная фигура конь* ♞ – код 9822 и др. Найти символ и соответствующий ему код можно на сайте **Таблица символов Юникода** (*unicode-table.com*). Но если выбранный вами символ отсутствует среди шрифтов на вашем компьютере, то он не будет отображаться в документе.

Идеограмма (древнегреч. ἰδέα – идея, γράμμα – письменный знак, буква) – письменный знак, передающий, в отличие от буквы, не звук какой-либо речи, а некоторое понятие, идею.

КОДИРОВАНИЕ СИМВОЛОВ ВЕБ-СТРАНИЦ

Особенности кодирования текста с использованием различных таблиц кодов символов можно увидеть также при просмотре веб-страниц. Иногда при открытии веб-страницы таблица кодов символов выбирается браузером неправильно. В таком случае текст веб-страницы непригоден для понимания (рис. 1.4).



Рис. 1.4. Вид веб-страницы при неправильном выборе таблицы кодов символов

У каждого веб-браузера есть инструменты для выбора таблицы кодов символов пользователями. Например, в браузере **Google Chrome** для изменения таблицы кодов символов следует:

1. Выбрать кнопку **Настройка и управление Google Chrome** ☰.
2. Выполнить ***Дополнительные инструменты*** ⇒ ***Кодировка***.
3. Выбрать таблицу кодов символов, при которой текст веб-страницы будет пригодным для понимания.

Для русскоязычных веб-страниц чаще всего применяют таблицу кодов символов **Юникод (UTF-8),** но для отдельных веб-страниц может быть использована кодировка **Кириллица (Windows-1251)** или **Кириллица (КОИ-8).**

Для тех, кто хочет знать больше

Таблица символов в операционной системе Windows

Среди служебных программ в операционной системе **Windows** есть программа **Таблица символов**, предназначенная для просмотра, поиска и вставки в текстовые документы разнообразных символов, в том числе отсутствующих на клавиатуре. Для запуска этой программы необходимо выполнить ***Пуск*** ⇒ ***Все программы*** ⇒ ***Стандартные*** ⇒ ***Служебные*** ⇒ ***Таблица символов***, после чего откроется окно программы (рис. 1.5).

Используя эту программу, можно увидеть различные группы символов, предназначенные для вставки в документы. Для этого необходимо:

1. Установить отметку флажка **Дополнительные параметры** в окне программы.
2. В списке **Набор символов** в нижней части окна выбрать ***Юникод***.
3. В списке **Группировка** выбрать ***Диапазоны Юникода***.
4. В списке открывшегося окна **Группировка** выбрать диапазон кодов, символы которого необходимо вставить в документ (рис. 1.6).



Рис. 1.5. Окно программы **Таблица символов**

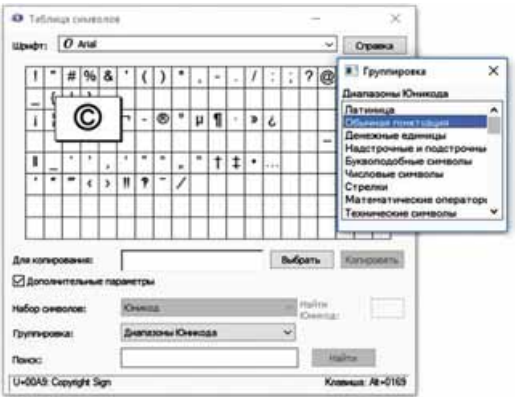


Рис. 1.6. Окно программы **Таблица символов** с символами группы **Обычная пунктуация**

Если в таблице выбрать некий символ, то в строке статуса окна программы слева отображается код этого символа в таблице кодов символов **Юникод** и его описание на английском языке, а справа – клавиши, которые следует нажать для вставки этого символа в документ. Например, на рисунке 1.6 выбран символ © – знак авторского права, в строке статуса окна приведено его описание – *U+00A9 Copyright Sign* и клавиши *Alt+0169* для вставки этого символа в документ.

Выбранные в таблице символы можно вставить в документы, создаваемые в различных программах – текстовом процессоре, редакторе презентаций, табличном процессоре и др. Для вставки символа в документ необходимо:

1. Выбрать символ в таблице с символами.
2. Выполнить ***Выбрать*** ⇒ ***Копировать***.
3. Выбрать место в документе.
4. Вставить скопированный символ из **Буфера обмена**.

После выбора кнопки **Выбрать** символ выводится в поле **Для копирования**. Если выбрать еще один символ и кнопку **Выбрать**, то в поле отобразятся оба символа. Таким образом можно подготовить для копирования несколько символов. Все вместе они будут скопированы и вставлены в документ.

Самое важное в этом пункте

Для обработки текстовых сообщений с использованием компьютера символы сообщения кодируют числами. Для этого могут использоваться разные таблицы кодов символов, например, **ASCII, КОИ-8, Windows-1251, Юникод.**

Таблица кодов символов **ASCII** содержит коды 128 символов, **КОИ-8, Windows-1251** – 256 символов, **Юникод** – 1 114 112 символов. Во всех таблицах кодов символов неизменными остаются первые 128 кодов, соответствующих таблице **ASCII.**

У каждого браузера есть инструменты для выбора таблицы кодов символов, с использованием которой будут отображаться символы на веб-странице.

Служебная программа **Таблица символов** операционной системы **Windows** предназначена для просмотра, поиска и вставки в документы символов из разных шрифтов, представленных с использованием различных таблиц кодов символов.

Дайте ответы на вопросы

1•. Как кодируют символы текстовых сообщений при их обработке с использованием компьютера?
2°. Какие таблицы кодов символов вы знаете?
3•. Какие символы могут быть закодированы с использованием таблиц кодов символов **ASCII, КОИ-8, Windows-1251, Юникод?**
4*. Что общего содержат и чем различаются таблицы кодов символов **ASCII, КОИ-8, Windows-1251, Юникод?**
5°. В каких ситуациях следует менять таблицу кодов символов при просмотре веб-страниц?
6•. Как выбрать таблицу кодов символов при просмотре веб-страниц в браузере **Google Chrome?**
7°. Каково назначение программы **Таблица символов?**
8°. Как запустить на выполнение программу **Таблица символов?**
9•. Как вставить в документ символы с использованием программы **Таблица символов?**

Выполните задания

1•. Сообщение закодировано с использованием таблицы кодов символов **ASCII.** Декодируйте его (воспользуйтесь приложением 1[1]):
а) 83 117 110; б) 104 111 109 101.

2•. Закодируйте сообщение с использованием таблицы кодов символов **Windows-1251** (воспользуйтесь приложением 2[1]):
а) *Учебник «Информатика»*; б) ваши фамилия и имя.

3*. Отдельные слова в сообщении закодированы с использованием таблицы кодов символов **Юникод.** Найдите символы по их кодам в

[1] Приложения можно найти на сайте **Інформатика для всіх** (allinf.at.ua) в разделе **Матеріали до підручників. 8 клас. Підручник.**

таблице и декодируйте сообщение. Запишите полученный текст. Для поиска символов можно использовать сайт **Таблица символов Юникода** (*unicode-table.com*).

Сегодня с утра 9925. Синоптики обещают 9928. Собираясь в 127979, не забудьте взять с собой 9730.

4•. Просмотрите веб-страницу **ИТ в Украине: истории и личности** *(ru.uacomputing.com)* с использованием таблиц кодов символов **Кириллица (Windows-1251)** и **Кириллица (КОИ-8)**. Установите, при выборе какой таблицы кодов символов текст веб-страницы пригоден для понимания.

5•. Вставьте в текстовый документ все символы, обозначающие денежные единицы, используя программу **Таблица символов**. Сохраните созданный документ в вашей папке в файле с именем **задание 1.2.5** и расширением имени, соответствующим версии текстового процессора.

6*. Вставьте в текстовый документ буквоподобные символы, используя программу **Таблица символов**. Сохраните созданный документ в вашей папке в файле с именем **задание 1.2.6** и расширением имени, соответствующим версии текстового процессора.

7*. Определите, какие символы будут вставлены в текстовый документ путем ввода их кода при нажатой клавише **Alt**, для кодов 1234, 3754, 8986, 9708, 9415, 10001, 10239.

1.3. ДВОИЧНОЕ КОДИРОВАНИЕ

1. В чем заключаются процессы кодирования и декодирования сообщений?
2. Какие таблицы кодов символов используют для кодирования текстовых сообщений?
3. Что означают приставки *кило*, *мега*, *гига*?

ДВОИЧНОЕ КОДИРОВАНИЕ. БИТ И БАЙТ



Рис. 1.7. Сэмюэл Морзе

Как вы уже знаете, для кодирования данных используют различные наборы сигналов. Для кодирования звуков речи используют буквы алфавита. В украинском языке букв 33, в английском – 26. Набор сигналов для кодирования числовых значений состоит из 10 цифр. Для кодирования команд, регулирующих порядок движения на пешеходных переходах, используют 3 цвета светофора. Наименьшее количество сигналов для кодирования сообщений имеет азбука Морзе – лишь 2 сигнала: *точка* и *тире*.

С 1844 года азбуку, предложенную американским художником Сэмюэлем Морзе (1791–1872) (рис. 1.7), впоследствии названную в его честь, ис-

пользовали для передачи сообщений по телеграфу. В азбуке Морзе буквы текстовых сообщений кодируются последовательностями точек и тире. Примеры кодирования букв русского алфавита приведены в таблице 1.3.

Таблица 1.3

Примеры кодирования букв в азбуке Морзе

Буква	*Значение кода в азбуке Морзе*
а	· –
о	– – –
п	· – – ·
р	· – ·
т	–

Например, слово *парта* кодируется с использованием азбуки Морзе последовательностью сигналов «· – – · · – · – · – · –».

Кодирование сообщений с использованием сигналов только двух видов называют **двоичным кодированием**. Сообщение, полученное в результате двоичного кодирования, называют **двоичным кодом** сообщения.

Двоичное кодирование используется и в современных компьютерах. В них сообщения кодируются последовательностями сигналов двух видов. Каждый сигнал одного вида условно обозначается цифрой **0**, а второго вида – цифрой **1**.

Цифру 0 или 1 в двоичном коде сообщения называют **бит** (англ. ***bi**nary digit* – двоичная цифра).

Из таблицы 1.3 видно, что в азбуке Морзе для кодирования буквы «т» используется 1 сигнал – тире, для кодирования буквы «а» – два, а для буквы «п» – четыре сигнала. Во время передачи сообщений по телеграфу одну букву от другой отделяют промежутками в подаче сигналов. Для обработки данных с использованием компьютера это не удобно. Проще отделить одну букву от другой, когда для их кодирования будет использовано одинаковое количество сигналов.

Если для кодирования сообщения используется 1 сигнал (бит) 0 или 1, то можно закодировать, например:

- вывод о правильности утверждения: *ложное* – 0 или *истинное* – 1;
- пол человека: *женский* – 0 или *мужской* – 1;
- состояние выключателя: *выключен* – 0 или *включен* – 1

и т. п.

Из двух битов можно составить 4 (2^2) разных кода (00, 01, 10 и 11). Ими можно закодировать, например, четыре стороны горизонта: 00 – север, 01 – восток, 10 – юг, 11 – запад.

Из трех битов можно составить уже 8 (2^3) разных кодов (000, 001, 010, 011, 100, 101, 110, 111). Ими можно закодировать, например, номера строк или столбцов на шахматной доске.

Из четырех битов можно составить $2^4 = 16$ кодов, из пяти – $2^5 = 32$ кода и т. д.

Из восьми битов можно составить $2^8 = 256$ кодов, и этого количества кодов достаточно, чтобы закодировать все буквы английского и русского (или какого-либо другого) алфавита, арабские цифры, знаки препинания, знаки арифметических действий, а также некоторые другие символы. Именно такое количество кодов содержится, например, в таблице кодов символов **Windows-1251**.

Последовательность из восьми битов называют **байт**.

1 байт = 8 бит.

ДЛИНА ДВОИЧНОГО КОДА СООБЩЕНИЙ

Если символ сообщения кодируется последовательностью из 8 бит, то длина двоичного кода этого символа равна 8 бит, или 1 байт.

Длина двоичного кода сообщения – это количество байтов в двоичном коде этого сообщения.

Например, сообщение *Информатика – интересный предмет!* содержит 33 символа (включая символы пробелов, тире, восклицательный знак). Если каждый символ кодировать двоичным кодом длиной 1 байт, то длина двоичного кода такого сообщения будет равна 33 байтам.

Как вы знаете, в таблице кодов символов **Windows-1251** каждому символу соответствует целое число от 0 до 255. Каждое из этих чисел может быть закодировано одним байтом. Примеры такого кодирования приведены в таблице 1.4.

Таблица 1.4

Примеры двоичного кодирования

Символ	*Числовой код символа*	*Соответствующий байт*
!	33	00100001
@	64	01000000
W	103	01100111
ю	254	11111110
я	255	11111111

Таким образом, каждому символу в таблице кодов символов **Windows-1251** соответствует двоичный код длиной 1 байт.

Чтобы выполнить двоичное кодирование текстового сообщения, можно каждый его символ заменить двоичным кодом, который соответствует числу из таблицы **Windows-1251**. Именно так кодируются текстовые документы, созданные в текстовом редакторе **Блокнот**, при их сохранении на носителе данных, если при сохранении используется кодировка **ANSI** (англ. *American National Standards Institute* – Американский национальный институт стандартов). Длина двоичного кода сообщения, содержащегося в таком документе, будет равняться в байтах количеству символов в текстовом документе. Следует учесть, что каждое нажатие клавиши **Enter** кодируется двумя специальными символами – символом перехода на новую строку и символом возвращения в начало строки.

Графические, звуковые, видеосообщения для их обработки компьютером также кодируются двоичными кодами.

Длину двоичного кода сообщений называют **объемом данных**.

КРАТНЫЕ ЕДИНИЦЫ ИЗМЕРЕНИЯ ДЛИНЫ ДВОИЧНОГО КОДА

Для обозначения длины двоичного кода сообщения используют кратные единицы измерения, образующиеся с использованием приставок *кило*, *мега*, *гига*, *тера* и др. Перечень этих приставок для обозначения кратных единиц измерения определен в Международной системе единиц (СИ). Там также приведены соответствующие множители. Но исторически сложилось так, что эти приставки (*кило, мега, гига, тера*) в информатике трактуются иначе, чем, например, в физике. С шестого класса вам известно, что:

1 Кбайт (килобайт)= 2^{10} байт = 1024 байт;

1 Мбайт(мегабайт) = 2^{10} Кбайт = 2^{20} байт = 1 048 576 байт;

1 Гбайт (гигабайт)= 2^{10} Мбайт = 2^{20} Кбайт = 2^{30} байт;

1 Тбайт(терабайт) = 2^{10} Гбайт = 2^{20} Мбайт = 2^{30} Кбайт = 2^{40} байт.

Так произошло потому, что в компьютере используются двоичные коды, для обработки которых удобно оперировать степенями числа 2, а не степенями числа 10. И поскольку 2^{10} = 1024, что приблизительно равно 1000, то именно 2^{10} байт = 1024 байт и стали называть **килобайт**. Аналогично, 2^{10} килобайт стали называть **мегабайт** и т. д.

ПЕРЕВОД ОДНИХ ЕДИНИЦ ИЗМЕРЕНИЯ ДЛИНЫ ДВОИЧНОГО КОДА В ДРУГИЕ

Длину двоичного кода сообщений можно указывать в битах, байтах и в кратных им единицах. Рассмотрим несколько задач на перевод одних единиц измерения в другие при определении длины двоичного кода сообщения.

Задача 1. Длина двоичного кода сообщения составляет 4,5 Мбайт. Выразить это значение в байтах.

Решение. Для перехода от мегабайтов к байтам сначала выразим длину двоичного кода этого сообщения в килобайтах. Учитываем, что 1 Мбайт = = 1024 Кбайт.

$$4{,}5 \text{ Мбайт} \cdot 1024 = 4608 \text{ Кбайт}.$$

Переведем килобайты в байты, учитывая, что 1 Кбайт = 1024 байт.

$$4608 \text{ Кбайт} \cdot 1024 = 4\,718\,592 \text{ байт}.$$

Ответ: 4,5 Мбайт = 4 718 592 байт.

Задача 2. Двоичный код сообщения состоит из 4 194 304 000 бит. Выразить длину двоичного кода этого сообщения целым числом в наибольших возможных единицах.

Решение. Переведем биты в байты, учитывая, что 1 байт = 8 бит.

$$4\,194\,304\,000 \text{ бит} : 8 = 524\,288\,000 \text{ байт}.$$

Полученное число больше множителя 1024, который используется для перехода от байтов к килобайтам в определении длины двоичного кода. Поэтому переведем байты в килобайты.

$$524\,288\,000 \text{ байт} : 1024 = 512\,000 \text{ Кбайт}.$$

Аналогично рассуждая, переведем килобайты в мегабайты.

512 000 Кбайт : 1024 = 500 Мбайт.

Полученное значение меньше чем 1024, поэтому перейти к большей единице с сохранением целого результата невозможно.

Ответ: 4 194 304 000 бит = 500 Мбайт.

Задача 3. Вычислить приблизительно длину двоичного кода текста учебника по информатике (не считая форматирования и рисунков).

Решение. Будем считать, что текст закодирован с использованием таблицы кодов символов **Windows-1251**, то есть длина двоичного кода каждого символа составляет 1 байт.

В каждой строке текста содержится около 60 символов, то есть длина двоичного кода строки составляет приблизительно 60 байт. На каждой странице размещено около 50 строк. Итак, длина двоичного кода страницы составляет примерно $60 \cdot 50 = 3000$ (байт). В учебнике около 250 страниц, то есть длина двоичного кода учебника составляет примерно $3000 \cdot 250 = 750\,000$ (байт).

Перейдем к более крупной единице измерения длины двоичного кода.

750 000 байт : 1024 ≈ 732,4 Кбайт.

Ответ: длина двоичного кода текста учебника по информатике примерно 732,4 Кбайт.

Самое важное в этом пункте

Кодирование сообщений с использованием сигналов только двух видов называют **двоичным кодированием.** Сообщение, полученное в результате двоичного кодирования исходного сообщения, называют **двоичным кодом** сообщения.

Цифру 0 или 1 в двоичном коде сообщения называют **бит.**

Последовательность из восьми битов называют **байт.**

1 байт = 8 бит.

Длина двоичного кода сообщения – это количество байт в двоичном коде этого сообщения.

Каждому символу в таблице кодов символов **Windows-1251** соответствует двоичный код длиной 1 байт.

1 Кбайт (килобайт) = 2^{10} байт = 1024 байт;

1 Мбайт (мегабайт) = 2^{10} Кбайт = 2^{20} байт = 1 048 576 байт;

1 Гбайт (гигабайт) = 2^{10} Мбайт = 2^{20} Кбайт = 2^{30} байт;

1 Тбайт (терабайт) = 2^{10} Гбайт = 2^{20} Мбайт = 2^{30} Кбайт = 2^{40} байт.

Дайте ответы на вопросы

1°. В чем заключается двоичное кодирование сообщений?

2°. Что такое 1 бит?

3°. Чему равен 1 байт?

4•. Что понимают под длиной двоичного кода сообщения?

5°. Какова длина двоичного кода каждого символа в таблице кодов символов **Windows-1251**?

6•. Какой будет длина двоичного кода сообщения, созданного в текстовом редакторе **Блокнот** и сохраненном в кодировке **ANSI**?

7*. На сколько будет отличаться длина двоичного кода сообщения, в котором указано ваше имя, если его создать в текстовом редакторе **Блокнот** и сохранить в кодировке **ANSI** в двух вариантах: записанным в одну строку и записанным в столбик по одной букве на строку?

8•. Чему равен 1 килобайт, 1 мегабайт, 1 гигабайт, 1 терабайт?

9•. Какую математическую операцию необходимо выполнить для перевода длины двоичного кода сообщений:
а) из битов в байты;
б) из байтов в килобайты;
в) из мегабайтов в килобайты?

Выполните задания

1°. Приведите примеры, где используется двоичное кодирование сообщений.

2•. Закодируйте с использованием азбуки Морзе слова *порт, рота, торт*. Сравните количество сигналов в двоичном коде этих слов.

3•. Определите длину двоичного кода сообщений при условии, что сообщения закодированы с использованием таблицы кодов символов **Windows-1251**:
а) *Таблица кодов символов Windows-1251*;
б) *В азбуке Морзе два символа – точка и тире.*

4•. Сравните длину двоичного кода текстового сообщения *Человек без друзей – что дерево без корней*, созданного в текстовом редакторе **Блокнот**, при сохранении его с использованием различных кодовых таблиц.

5•. Заполните пропуски:
а) 5 байт = _____ бит
б) 4096 байт = _____ Кбайт
в) 10 Кбайт = _____ бит
г) 3 Мбайт = _____ байт

6•. Заполните пропуски:
а) 2 байт = _____ бит
б) 8192 байт = _____ Кбайт
в) 2 Кбайт = _____ бит
г) 100 Мбайт = _____ Кбайт

7*. В одной строке некоторого текста размещается в среднем 60 символов, а на одной странице – 40 строк. Сколько закодированных страниц этого текста можно взять, чтобы длина двоичного кода не превысила 8 Мбайт, если для кодирования использовать таблицу **Windows-1251**?

ПРАКТИЧЕСКАЯ РАБОТА № 1

«Решение задач на определение длины двоичного кода текстовых данных»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Вычислите длину двоичного кода некоторого текста при условии, что символы текста закодированы с использованием таблицы кодов символов **Windows-1251.** В тексте 32 страницы, каждая страница содержит 48 строк, в каждой строке в среднем 56 символов. Выразите длину двоичного кода этого текста целым числом в наибольших возможных единицах.
2. Определите, используя текстовый редактор **Блокнот,** длину двоичного кода сообщения *Практическая работа № 1* при условии, что сообщение закодировано с использованием таблицы кодов символов:
   а) **Windows-1251;**
   б) **Юникод.**
   Сохраните текстовые документы с сообщением в файлах с именами **практическая 1 Windows.txt** и **практическая 1 Юникод.txt.**
3. Заполните в табличном процессоре таблицу:

Бит	***Байт***	***Килобайт***	***Мегабайт***
614 400			
	524 288		
		256	
			4

Сохраните результат в файле с именем **практическая 1.xlsx.**

Глава 2. Аппаратно-программное обеспечение компьютера

В этой главе вы узнаете:

- об архитектуре компьютера
- об истории средств обработки сообщений
- о классификации и общей характеристике программного обеспечения
- о системном и служебном программном обеспечении
- о классификации и основных функциях операционных систем
- об архивировании и сжатии данных
- об устройствах компьютера, их назначении и основных характеристиках:
  - процессоров
  - внешних и внутренних запоминающих устройств
  - мониторов
  - принтеров
  - мультимедийного оборудования



2.1. АРХИТЕКТУРА КОМПЬЮТЕРА

1. Какие устройства входят в состав компьютера? Для чего они предназначены?
2. Какое устройство компьютера осуществляет обработку данных? Где, как правило, оно размещается?
3. Назовите информационные процессы. Какие устройства компьютера обеспечивают хранение данных?

АРХИТЕКТУРА КОМПЬЮТЕРА

В пятом классе вы уже ознакомились с устройствами персонального компьютера. Рассмотрим несколько подробнее схему реализации информационных процессов в компьютере с использованием его устройств (рис. 2.1). Данные попадают во внутренние запоминающие устройства (внутреннюю память) компьютера через устройства ввода данных (клави-

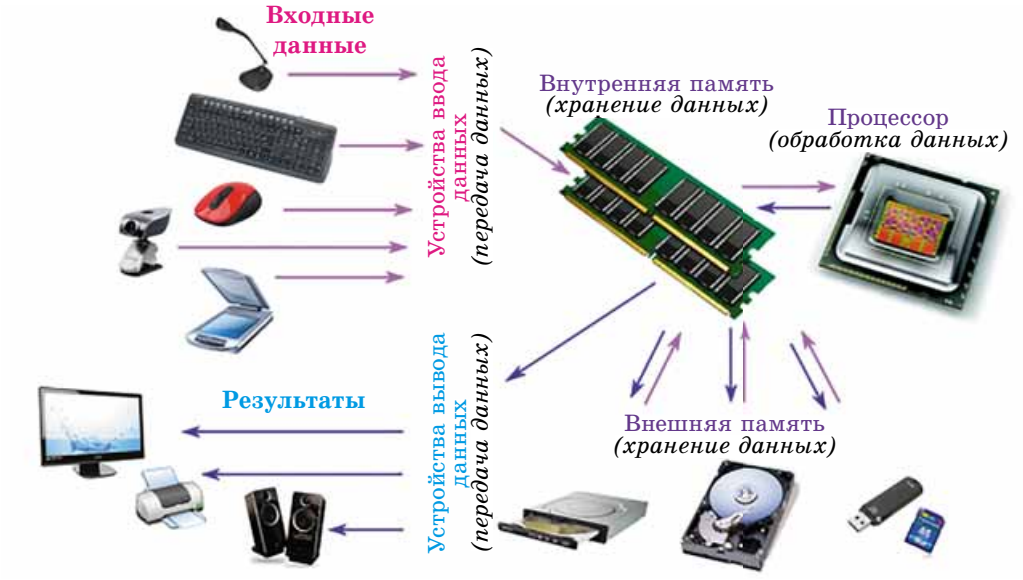


Рис. 2.1. Схема реализации информационных процессов в персональном компьютере.

атуру, микрофон, мышь, сканер и т. п.) или с внешних запоминающих устройств (внешней памяти) – устройств для работы с жесткими магнитными и оптическими дисками, флэш-накопителей, флэш-карт и т. п. Из внутренней памяти данные попадают в процессор, где обрабатываются, и результаты обработки снова передаются во внутреннюю память, а оттуда – к устройствам вывода данных (монитор, принтер, звуковые колонки и т. п.) или к устройствам внешней памяти. Все операции в компьютере, связанные с реализацией информационных процессов, возможны только под управлением соответствующих программ.

Архитектура (греч. αρχιτεκτόνημα) – здание, замысел, затея.

Схема, изображенная на рисунке 2.1, описывает типичную архитектуру персонального компьютера. **Архитектура компьютера** – это модель, описывающая взаимодействие устройств и программ для обеспечения информационных процессов в компьютере.

ПРОЦЕССОР, ЕГО НАЗНАЧЕНИЕ

Вы уже знаете, что обработка данных в компьютере выполняется процессором (рис. 2.2). Он представляет собой устройство, обеспечивающее выполнение компьютерных программ. У процессора, как у любого исполнителя алгоритмов, есть свой набор команд.



Рис. 2.2. **Процессор Helio X20** с 10-ю ядрами

Процессор компьютера является его основной составляющей. Он содержит **устройство управления**, которое обеспечивает выполнение команд компьютерной программы, и **арифметико-логическое устройство**, которое и осуществляет операции над данными. В состав современных процессоров входит еще и составляющая внутренней памяти – так называемая **кэш-память**. Ее назначение мы рассмотрим немного позже.

Основным свойством процессора является ***скорость обработки данных***, которая зависит от значения нескольких других свойств: **тактовой частоты процессора, количества ядер, разрядности, объемов кэш-памяти** и т. п. (табл. 2.1).

Таблица 2.1

Примеры значений свойств процессора для настольного и планшетного компьютеров

Свойство	***Что характеризует***	***Значение свойств процессора***	
		для настольного компьютера	*для планшетного компьютера*
Модель	Имя, присваиваемое фирмой-производителем	Intel Core i5–6600	Qualcomm Snapdragon MSM8939
Тактовая частота	Частота управляющих сигналов, которые согласовывают работу устройств процессора	3,3 ГГц	1,8 ГГц
Количество ядер	Количество одинаковых по структуре процессоров, объединенных в одно устройство	4	8
Разрядность	Количество двоичных разрядов, которые могут одновременно обрабатываться процессором	64 бит	64 бит

ПАМЯТЬ КОМПЬЮТЕРА. ВНУТРЕННЯЯ ПАМЯТЬ

Память компьютера предназначена для хранения данных. Ее разделяют на **внутреннюю** и **внешнюю** (рис. 2.3). Основным свойством всех видов памяти является ее емкость. Она измеряется в байтах, килобайтах, мегабайтах, гигабайтах, терабайтах и т. п.

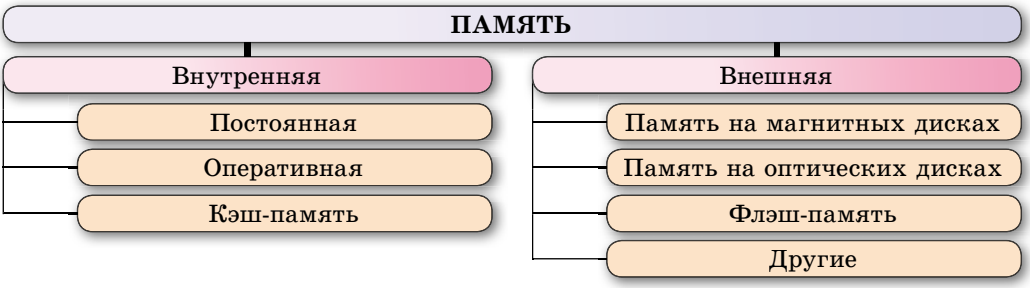
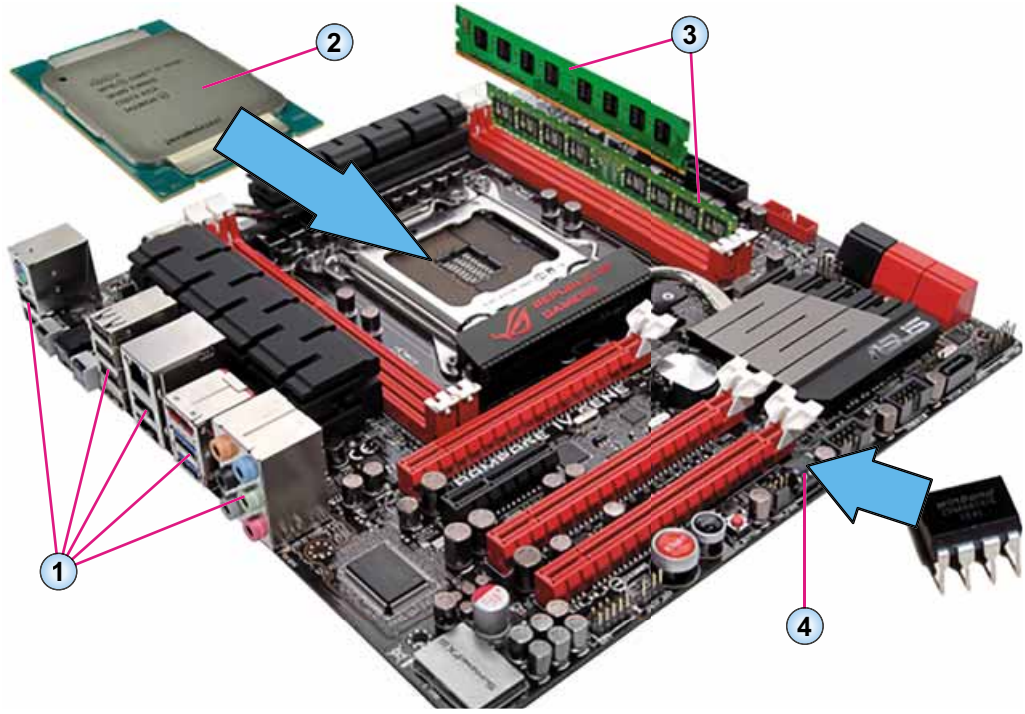


Рис. 2.3. Виды памяти

Материнская, или **главная, плата** (англ. *motherboard* или *mainboard*) – основная составляющая современных персональных компьютеров, которая обеспечивает передачу данных между устройствами компьютера.

Внутренняя память служит для обеспечения работы процессора. Устройства внутренней памяти, как и процессор, размещаются на материнской плате компьютера (рис. 2.4).

Оперативная память является основной в компьютере, в ней размещаются программы и данные, которые в даль-



1. Разъемы для подключения устройств ввода и вывода данных
2. Процессор
3. Блоки микросхем оперативной памяти
4. Микросхема постоянной памяти

Рис. 2.4. Материнская плата компьютера

нейшем обрабатываются процессором. Он может получать данные и команды только из оперативной памяти. В нее же записываются и результаты обработки данных. Для того чтобы данные были обработаны процессором, они предварительно должны быть переданы («загружены») из устройств ввода данных или запоминающих устройств в оперативную память.

После выключения компьютера все данные с оперативной памяти исчезают.

Слот (англ. *slot*) – щель, паз, отверстие.
Сокет (англ. *socket*) – углубление, гнездо, розетка.
Кэш (англ. *cache*) – тайник, запас пищи.

Оперативная память изготавливается в виде блока микросхем, который устанавливается в специальные разъемы (слоты) на материнской плате (рис. 2.4, 3). Как правило, на материнской плате можно установить несколько блоков микросхем оперативной памяти.

а) 

б) 

Рис. 2.5. Микросхемы постоянной памяти на материнской плате

Для обеспечения начала работы компьютера основные (базовые) программы хранятся в **постоянной памяти.** При включении компьютера данные из постоянной памяти копируются («загружаются») в оперативную память, и запускаются программы подготовки компьютера к работе.

Постоянная память изготавливается в виде одной или двух микросхем, которые вставляются в специальные гнезда – сокеты (рис. 2.5, *а*) или впаиваются в материнскую плату (рис. 2.5, *б*).

Как уже отмечалось, в состав современных процессоров включают **кэш-память.** Она предназначена для ускорения обмена данными между основным массивом оперативной памяти и процессором. Различают кэш-память первого, второго и третьего уровней. Емкость кэш-памяти первого уровня в большинстве процессоров – 128 Кбайт. Емкости кэш-памяти второго и третьего уровней существенно отличаются в разных процессорах. На начало 2016 года кэш-память второго уровня в процессорах для настольных персональных компьютеров составляет 1–8 Мбайт, а третьего – 2–20 Мбайт. Увеличение емкости кэш-памяти обычно увеличивает быстродействие процессора.

ВНЕШНИЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА

Если устройства внутренней памяти размещены на материнской плате, то устройства внешней памяти присоединяются к ней с использованием различных соединений. Устройства внешней памяти, в отличие от устройств внутренней памяти (за исключением постоянной памяти), пред-

назначены для длительного хранения данных. После выключения питания данные с носителей внешней памяти не исчезают.

К устройствам внешней памяти относятся накопители на жестких магнитных дисках, устройства для работы с оптическими дисками, устройства, использующие флэш-память и т. п.



1. Поверхность одного из жестких дисков
2. Электромагнитная головка

Рис. 2.6. Устройство накопителя на жестких магнитных дисках

Носителем данных в **накопителе на жестких магнитных дисках** (НЖМД) является один или несколько металлических дисков, покрытых слоем магнитного вещества (рис. 2.6). Запись данных производится путем намагничивания участков поверхности диска с использованием электромагнитной головки, которая выполняет и считывание данных.

Свойства накопителей на жестких магнитных дисках и примеры их значений в современных устройствах представлены в таблице 2.2.

Таблица 2.2

Значение свойств современных НЖМД

Свойство	***Значение свойства НЖМД***	
	для настольных компьютеров	*для ноутбуков*
Емкость	6 Тбайт	2000 Гбайт
Диаметр жестких дисков	3,5 дюйма	2,5 дюйма
Скорость вращения жестких дисков	5400–7200 об/мин	5400 об/мин

Устройства для работы с **оптическими дисками** используют луч лазера для считывания (записи) данных с оптических дисков различных типов, которые отличаются емкостью и возможностью выполнять запись и перезапись данных. Основные типы оптических дисков, их свойства и значения свойств представлены в таблице 2.3.

Таблица 2.3

Основные типы оптических дисков

Обозначение дисков, предназначенных для			***Емкость дисков***
только считывания данных	*считывания и записи данных*	*считывания, записи и перезаписи данных*	
CD-ROM	CD-R	CD-RW	640–800 Мбайт
DVD-ROM	DVD-R, DVD+R	DVD-RW DVD+RW	9,4 Гбайт
BD-ROM	BD-R	BD-RE	100 Гбайт

Устройства флэш-памяти в последнее время получили широкое распространение и используются не только как устройства для переноса данных между компьютерами, но и как основное устройство внешней памяти в переносных компьютерах – нетбуках, планшетных компьютерах, не говоря уже о мобильных устройствах, а также фото- и видеокамерах (рис. 2.7). Флэш-память значительно уменьшила использование оптических дисков.



Рис. 2.7. Флэш-карты разных типов



Рис. 2.8. SSD-диск

SSD (англ. ***Solid-State Drive***) – твердотелое (сплошное) устройство.

С увеличением емкости устройств флэш-памяти они начали активно использоваться и как заменители накопителей на жестких магнитных дисках. Такие устройства получили название **твердотелых накопителей,** или **SSD-дисков.** Емкость SSD-дисков (рис. 2.8) не намного меньше емкости накопителей на жестких магнитных дисках и составляет более чем 2 Тбайт, однако они могут обеспечивать большую скорость записи и считывания данных.

Самое важное в этом пункте

Архитектура компьютера – это модель, описывающая взаимодействие устройств и программ для обеспечения информационных процессов в компьютере.

Процессор – это устройство, обеспечивающее обработку данных. Он содержит **устройство управления**, которое обеспечивает выполнение команд компьютерной программы, и **арифметико-логическое устройство**, которое и осуществляет операции над данными. Основным свойством процессора является ***скорость обработки данных***, зависящая от значения нескольких других свойств: **тактовой частоты**, **количества ядер**, **разрядности**, **объемов кэш-памяти** и т. п.

Память компьютера предназначена для сохранения данных, ее разделяют на **внутреннюю** и **внешнюю.**

Устройства **внутренней памяти** (оперативная, постоянная, кэш-память) размещаются на материнской плате. Данные из оперативной памяти и кэш-памяти после выключения питания исчезают, а с постоянной – нет.

Устройства **внешней памяти** (накопители на жестких магнитных дисках, устройства для работы с оптическими дисками, устройства, использующие флэш-память) предназначены для длительного хранения данных. После выключения питания данные с носителей внешней памяти не исчезают.

Основным свойством всех видов памяти является ее емкость, которая измеряется в байтах, килобайтах, мегабайтах, гигабайтах, терабайтах и т. п.

Дайте ответы на вопросы

1°. Что такое архитектура компьютера?
2•. Опишите схему реализации информационных процессов в компьютере по рисунку 2.1.
3•. Для чего предназначен процессор? Назовите устройства, входящие в состав процессора.
4°. Какое устройство памяти включено в состав современных процессоров?
5•. Назовите свойства процессора и приведите примеры их значений.
6•. Опишите классификацию памяти компьютера. Каково основное свойство памяти?
7°. Назовите единицы измерения емкости памяти.
8•. Для чего предназначена оперативная память? Где она размещается?
9•. Для чего предназначена постоянная память компьютера? Чем она отличается от оперативной?
10°. Для чего предназначена кэш-память?
11•. Назовите устройства внешней памяти. Приведите примеры значений свойств каждого из них.
12•. Чем внутренняя память отличается от внешней?
13*. Что общего и в чем отличие между устройствами внешней памяти?

Выполните задания

1•. Используя данные одного из сайтов со сведениями о современных компьютерных устройствах, например **hotline.ua**, определите значения свойств трех различных процессоров, поступивших в последнее время в продажу в магазины Украины. Заполните таблицу:

Свойства	*Значение свойств процессора*		
Модель			
Тактовая частота			
Количество ядер			
Разрядность			

2*. Исследуйте, используя данные сайтов со сведениями о современных компьютерных устройствах, например **hotline.ua**, значения свойств блоков оперативной памяти максимальной емкости, предлагаемых для ноутбуков. Заполните таблицу:

Свойства	*Значение свойств*		
Модель			
Производитель			
Емкость			
Стоимость			

3•. Расположите носители данных, которые вы знаете, в порядке убывания их емкости.

4•. По сведениям из Интернета или каталогам компьютерных магазинов определите значения свойств процессоров, которые доступны на рынке компьютеров вашего региона.

5*. По сведениям из Интернета или каталогам компьютерных магазинов сравните емкость кэш-памяти второго уровня процессоров с одинаковыми значениями тактовой частоты и количества ядер различных производителей.

6*. Подготовьте сообщение о возможности и способах изменения данных в постоянной памяти.

2.2. УСТРОЙСТВА ВВОДА И ВЫВОДА ДАННЫХ

1. Какие устройства ввода данных входят в состав компьютера? Для ввода каких данных они используются?
2. Какие устройства вывода данных входят в состав компьютера? Для вывода каких данных они используются?
3. Какие устройства используются для ввода и вывода мультимедийных данных? Приведите примеры таких устройств.

УСТРОЙСТВА ВВОДА ДАННЫХ

Как вы уже знаете, в состав компьютеров входят устройства ввода и вывода данных.



Рис. 2.9. Геймпад



Рис. 2.10. Кейпад



Рис. 2.11. Танцевальная площадка

Устройства ввода данных можно разделить по типу данных, с которыми они работают. Так, можно выделить устройства ввода текстовых **(клавиатура)**, графических **(сканер, фотокамера, графический планшет)**, звуковых **(микрофон)**, видеоданных **(видеокамера, веб-камера, ТВ-тюнер)**. Еще одной группой устройств ввода данных являются устройства для обеспечения управления в различных программах. К ним относятся **мышь, тачпад, мультимедийная (электронная) доска, сенсорный экран, джойстик, геймпад** (рис. 2.9), **кейпад** (рис. 2.10), **руль, педали, танцевальная площадка (дэнспад)** (рис. 2.11) и т. п.

Геймпад (англ. *gamepad*) – игровая площадка (площадка, подкладка).
Кейпад (англ. *keypad*) – площадка для клавиш.
Дэнспад (англ. *dancepad*) – площадка для танцев.

В компьютерах, которые используются для учебных и научных целей, могут применяться дополнительные устройства для ввода данных. Например, датчики для определения значения температуры, влажности воздуха, наличия определенных примесей в веществах, скорости движения различных объектов и т. п.

Рассмотрим свойства (технические характеристики) и классификацию устройств ввода данных.

Клавиатура предназначена для ввода символьных данных и команд. Несмотря на то что клавиатуры за последние несколько десятилетий очень мало изменялись, их можно разделить на несколько групп в зависимости от значения определенных свойств:

- по назначению: *стандартные* (для настольных ПК), *компактные*, *мультимедийные*, *игровые* и т. п.;
- по типу подключения к системному блоку: *проводные* или *беспроводные* (*радио*, *Bluetooth*, *Wi-Fi* и т. п.).

Для переносных компьютеров могут быть использованы проекционные клавиатуры (рис. 2.12).



Рис. 2.12. Проекционная клавиатура

Манипуляторы **мышь** делятся в зависимости от значений следующих свойств:

- по назначению: *для настольных ПК*, *для ноутбуков* (*компактные*), *игровые* и т. п.;
- по типу подключения к системному блоку: *проводные* или *беспроводные* (*радио*, *Bluetooth*, *Wi-Fi* и т. п.);
- по типу датчиков движения: *оптические*, *лазерные*;
- по количеству кнопок: *2*, *3–5*, *6–9*, *10 и более* и т. п.;
- по типу корпуса: *симметричная стандартная*, *симметричная для левши*, *эргономичная* и т. п.



Рис. 2.13. Мышь

На рисунке 2.13 изображена игровая проводная лазерная эргономичная мышь, имеющая 19 кнопок и одно колесико.

Тачпады (англ. *touch* – прикосновение, *pad* – площадка, подушечка) встраиваются практически во все типы ноутбуков и нетбуков, однако могут использоваться и при работе с настольными ПК (рис. 2.14). Перемещение пальца по поверхности тачпада приводит к перемещению указателя на экране монитора. Однократное кратковременное нажатие на площадку тачпада аналогично однократному щелчку ле-



Рис. 2.14. Внешний тачпад

вой кнопкой мыши, а двойное нажатие – двойному щелчку левой кнопкой мыши. Левая и правая кнопки тачпада выполняют те же функции, что и соответствующие кнопки мыши.

Мультимедийные (электронные) доски используют в основном в учебных заведениях, а также при проведении различных презентаций. Различают доски в зависимости от их размеров (длина диагонали *70–79*, *80–89* и более *90 дюймов*) и от принципа определения места прикосновения к доске.

Сенсор (англ. *sensor* от лат. *sensus* – ощущение) – датчик, устройство, которое превращает сигнал в форму, удобную для обработки.
Стилус (лат. *stilus*) – палочка для письма.

Сенсорные экраны используют в основном в планшетных компьютерах, смартфонах, а также в различных справочных (рис. 2.15) и презентационных системах. Ввод данных происходит прикосновением пальца или стилуса к определенным участкам поверхности экрана. Экран «чувствует», в каком месте произошло прикосновение, и передает соответствующий сигнал компьютеру. Основные свойства – размер экрана (длина диагонали может быть *от 3 до 70 и более дюймов*) и принцип отслеживания места прикосновения.

Близкий по принципу действия к сенсорному экрану **графический планшет**. Он используется для создания рисунков и ввода других графических данных. Пользователь создает рисунок, касаясь стилусом поверхности графического планшета. В отдельных видах графических планшетов рабочая поверхность не отражает изображение, оно появляется на экране компьютера, к которому подключен графический планшет. В других созданное изображение отображается на рабочей поверхности графического планшета (рис. 2.16).



Рис. 2.15. Пункт информационно-справочной системы железной дороги

Различают графические планшеты по размеру рабочей поверхности, разрешению (*от 2000 до более чем 4000 точек на дюйм*), а также по количеству вариантов восприятия силы нажатия на рабочую поверхность (*от 512 до 2048*).

Сканеры также используют для ввода графических данных. Среди них по принципу действия выделяют *планшетные*, *с протяжным механизмом*, *ручные* и *скан-камеры* (рис. 2.17). Основными свойствами сканеров являются размер области сканирования, разрешение (*от 600 до более чем 6400 точек на дюйм*) и скорость сканирования (*5–20 с*).



Рис. 2.16. Графический планшет

С использованием фото- и видеокамер, веб-камер, микрофонов и других мультимедийных устройств вы познакомились в курсе информатики шестого класса. Отметим, что основным свойством фото- и видеокамер, веб-камер явля-

ется их разрешение, которое определяет качество изображения. Для **видеокамер** качество изображения определяется стандартами **SD** (англ. *Standard Definition* – стандартная четкость, *720 × 576* точек), **HD** (англ. *High Definition* – высокая четкость, *1280 × 720* точек), **Full HD** (англ. *Full High Definition* – полная высокая четкость, *1920 × 1080* точек), **Ultra HD** (англ. *Ultra High Definition* – сверхвысокая четкость, *3840 × 2160* точек).



планшетный | с протяжным механизмом | скан-камера

Рис. 2.17. Типы сканеров

В **фотокамерах** качество изображения зависит от оптических свойств объектива и количества точек (пикселей) (*от 10 до более 24 мегапикселей*), которое может воспроизвести устройство. **Веб-камеры** имеют разрешение *от 320 × 240 пикселей до 2560 × 2048 пикселей* и более.

ТВ-тюнер – это устройство, обеспечивающее просмотр телевизионных передач на компьютере.

УСТРОЙСТВА ВЫВОДА ДАННЫХ

Устройства вывода, как и устройства ввода данных, можно разделить по типу данных, с которыми они работают. Так, можно выделить устройства вывода текстовых и графических данных (**монитор, принтер, плоттер**), звуковых (**наушники, звуковые колонки**), видеоданных (**мультимедийные проекторы, экранные панели**). Некоторые устройства могут выводить как графические, так и видеоданные (монитор, мультимедийные проекторы, экранные панели и т. п.).

Монитор является основным устройством для вывода данных в персональных компьютерах. Их разделяют по способу создания изображения на ***мониторы на жидких кристаллах,*** или **LCD**-мониторы (англ. ***Liquid Crystal Display*** – жидкокристаллический дисплей), ***плазменные***, ***OLED*** (англ. ***Organic Light Emitting Diode*** – органический светодиод), на ***электронных чернилах*** – ***e-ink*** (англ. *electronic* ***ink*** – электронные чернила) и т. п. В большинстве современных мониторов используется жидкокристаллическая технология.

LCD-мониторы разделяют по принципу действия (*TN + film, PLS,* различные версии *IPS* и *VA*), по длине диагонали (*от 19 до более 30 дюймов*), разрешению (*от 1280 × 1024 до 5120 × 2880 пикселей*), по времени реакции – отклика на команду смены изображения на экране (*от 1 до 8 мс*).

Для обработки графических данных, выводящихся на экран монитора, в компьютерах используется специальное устройство – **видеоадаптер**. Различают интегрированные видеоадаптеры и в виде отдельной платы. В случае интегрированного видеоадаптера он размещается в микросхеме процессора или на материнской плате. Интегрированные видеоадаптеры используются в персональных компьютерах, которые не связаны с качественной обработкой больших объемов графических данных. Для этих целей используют видеоадаптеры в виде отдельной платы (рис. 2.18), которая вставляется в один из слотов материнской платы. Видеоадаптер содержит специальный графический процессор и дополнительную оперативную память – графическую (видео) память. Различают графические адаптеры по типу процессора (например, *Radeon R7 370*, *GeForce GTX 950*), объему памяти (от *1* до *6 Гбайт* и более), системе подключения к материнской плате и к монитору и т. п.

Интегральный (лат. *integer* – целый) – неразрывно связанный, цельный.
Интеграция (лат. *integratio* – пополнение) – объединение в целое отдельных частей.
Чипсет (англ. *chip* – микросхема, *set* – упорядочивать) – набор микросхем, обеспечивающий обмен данными между устройствами компьютера.



Рис. 2.18. Видеоадаптер

Для вывода данных на бумагу, пленку или другие материалы используются ***принтеры*** и ***плоттеры***.

Принтеры разделяют по принципу действия на *матричные*, *лазерные*, *струйные*, *термические*, *3D-принтеры* и т. п. Основными свойствами принтеров являются:

- размер материала, на который выполняется печать (А4, А3 и т. п.);
- количество цветов – *монохромные*, *многоцветные*;
- разрешение – от *600 × 1200 точек* до *5760 × 1440 точек* на дюйм и более;
- скорость печати – от *1* до *50 и более* страниц в минуту;
- тип разъемов для подключения к компьютеру – *проводное* или *беспроводное*.

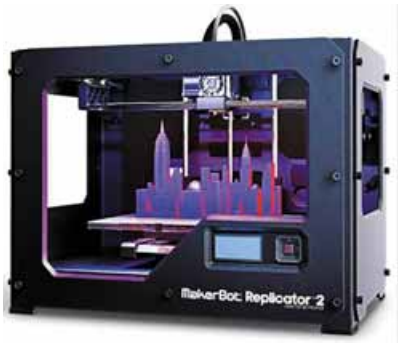


Рис. 2.19. 3D-принтер

В последнее время все большую популярность приобретают ***3D-принтеры***, обеспечивающие создание изделий сложной формы (рис. 2.19).

УСТРОЙСТВА, ВХОДЯЩИЕ В СОСТАВ МУЛЬТИМЕДИЙНОГО ОБОРУДОВАНИЯ

Вы уже знаете, что среди устройств ввода и вывода можно выделить устройства, которые работают с мультимедийными данными. Напомним, что к таким устройствам относятся **фотокамера**, **графический планшет**,

микрофон, видеокамера, веб-камера, мультимедийные (электронные) доски, наушники, звуковые колонки, мультимедийные проекторы, экранные панели и т. п. Для устройств, работающих с неподвижными и движущимися изображениями, одним из основных свойств является разрешение, а для работающих со звуком – полоса частот звукового диапазона, который может обработать устройство.

Самое важное в этом пункте

К устройствам ввода данных относятся устройства ввода текстовых **(клавиатура)**, графических **(сканер, фотокамера, графический планшет)**, звуковых **(микрофон)**, видеоданных **(видеокамера, веб-камера, ТВ-тюнер)**, а также устройства для обеспечения управления объектами в различных программах – **мышь, тачпад, мультимедийная (электронная) доска, сенсорный экран, джойстик, геймпад, кейпад, руль, педали, танцевальная площадка** и т. п.

К устройствам вывода данных относятся устройства вывода текстовых и графических данных **(монитор, принтер, плоттер)**, звуковых **(наушники, звуковые колонки)**, видеоданных **(мультимедийные проекторы, экранные панели)** и т. п.

Классифицируют указанные устройства по значению определенных свойств.

Дайте ответы на вопросы

1°. Какие устройства компьютера принадлежат к устройствам ввода данных?

2°. Какие устройства компьютера принадлежат к устройствам вывода данных?

3•. По значению каких свойств классифицируют клавиатуры? Приведите примеры.

4•. Какие виды манипуляторов мышь используются в компьютерном классе вашей школы? Назовите значения их свойств.

5*. Какие значения свойств могут иметь сканеры, используемые в компьютерных классах школы?

6•. Назовите свойства монитора компьютера и приведите примеры их значений.

7*. Опишите назначение монитора и видеоадаптера. Какие значения свойств могут иметь современные видеоадаптеры?

8•. Опишите классификацию современных принтеров. Какие виды принтеров используются в вашей школе?

9*. Какие значения свойств должен, по вашему мнению, иметь принтер для домашнего пользования?

10•. Для чего предназначены устройства, входящие в состав мультимедийного оборудования?

11*. Укажите, что общего в графических планшетах и планшетных компьютерах и чем они отличаются. Для каких целей используются первые и вторые?

Выполните задания

1•. Используя сведения, размещенные в учебнике, подготовьте сообщение о классификации устройств ввода данных.

2•. Используя сведения, размещенные в учебнике, подготовьте слайд презентации со схемой классификации устройств ввода данных. Сохраните презентацию в файле с именем **задание 2.2.2** в вашей папке.

3•. Используя данные одного из сайтов со сведениями о современных компьютерных устройствах, например **hotline.ua**, определите значения свойств трех различных манипуляторов мышь, поступивших в последнее время в продажу в магазины Украины. Заполните таблицу:

Свойства	*Значения свойств мыши*		
Модель			
Тип подключения			
Тип датчиков движения			
Количество кнопок			
Тип корпуса			

4*. Используя данные сайтов со сведениями о современных компьютерных устройствах, например **hotline.ua**, определите значения свойств мониторов трех различных производителей, предлагаемых магазинами Украины, заполните таблицу:

Свойства	*Значения свойств*		
Модель			
Производитель			
Длина диагонали, *дюймов*			
Тип матрицы			
Максимальное разрешение, *пикселей*			
Час реакции, *мс*			

5•. Используя сведения, размещенные в учебнике, подготовьте в одной из прикладных программ схему классификации современных принтеров. Сохраните схему в файле с именем **задание 2.2.5** в вашей папке.

6•. Определите значения свойств устройств вашего домашнего компьютера и заполните таблицу:

Свойства	*Значения свойств*		
Тип видеоадаптера			
Объем графической (видео) памяти, *байт*			
Тип клавиатуры			
Тип мыши			

7•. Используя сведения из Интернета, каталогов компьютерных магазинов, определите значения свойств принтеров трех различных производителей, предлагаемых магазинами Украины. Заполните таблицу:

Свойства	*Значения свойств*		
Модель			
Производитель			
Принцип действия			
Разрешение, *пикселей*			
Скорость печати, *страниц в минуту*			

8*. Подготовьте сообщение об одном из новых видов устройств ввода данных – **кейпаде**. Опишите его устройство, назначение и область применения.

9*. Подготовьте сообщение об одном из новых видов устройств ввода данных – **дэнспаде**. Опишите его устройство, назначение и область применения.

2.3. ИСТОРИЯ СРЕДСТВ РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ. ВИДЫ СОВРЕМЕННЫХ КОМПЬЮТЕРОВ

1. Какие виды компьютеров вы знаете? Чем отличается их использование?
2. Какие вам известны примеры применения компьютеров в различных отраслях человеческой деятельности?
3. Назовите фамилии украинских ученых, которые внесли весомый вклад в развитие компьютерной техники.

ЭТАПЫ РАЗВИТИЯ СРЕДСТВ РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

Развитие человечества непосредственно связано с развитием средств передачи, обработки и хранения сообщений. В этом развитии можно выделить несколько этапов (табл. 2.4).

Таблица 2.4

Этапы развития средств реализации информационных процессов

Название этапа	*Период истории человечества*	*Примеры носителей данных, средств передачи и обработки данных*
Этап ручных средств	От древних времен до середины XV в.	*Носители данных* – глиняные таблички, папирус, береста, палки с зарубками, картины. *Средства передачи* – сигнальные костры, барабанный бой, персональные посланцы, голубиная почта, первые почтовые службы для передачи государственных документов. *Средства обработки* – мозг и пальцы человека, ручные вычислительные приборы (абак, счеты и т. п.).

Продолжение таблицы 2.4

Название этапа	***Период истории человечества***	***Примеры носителей данных, средств передачи и обработки данных***
Этап механических средств	Со средины XV в. до середины XIX в.	*Носители данных* – книги, газеты, журналы, фотопластинки. *Средства передачи* – почтовые службы. *Средства обработки* – пишущие машинки, фотоаппараты, арифмометры, машинки для вычислений.
Этап электрических средств	Со средины XIX в. до 40-х годов XX в.	*Носители данных* – пластинки, кинопленки, магнитные пленки. *Средства передачи* – телефон, телеграф, радио. *Средства обработки* – фонографы, граммофоны, электрические пишущие машинки, табуляторы, электрические арифмометры, магнитофоны, ротапринт.
Этап электронных средств	От 40-х годов XX в. до наших дней	*Носители данных* – магнитные и оптические диски, электронные микросхемы. *Средства передачи* – телевидение, компьютерные сети, средства мобильной связи. *Средства обработки* – электронные калькуляторы, компьютер.

ИСТОРИЯ СРЕДСТВ ОБРАБОТКИ СООБЩЕНИЙ

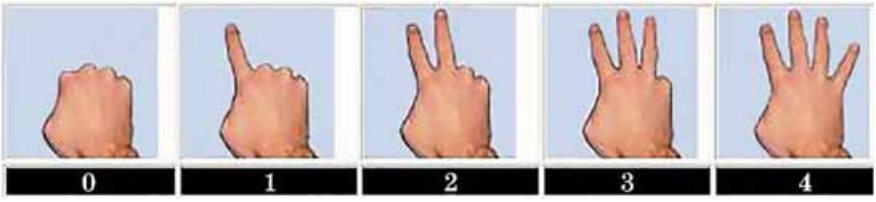


Рис. 2.20. Счет с помощью пальцев руки

Рассмотрим подробнее историю средств обработки сообщений, предназначенных для выполнения вычислений. Первыми средствами обработки числовых сообщений были пальцы (рис. 2.20). Об этом свидетельствуют, например, римские цифры (I, V, X). В древней славянской нумерации единицы называли «перстами», то есть пальцами, а название цифры 5 происходит от слова «пясть» (кисть руки).

Значительным шагом вперед в развитии средств вычислений стало создание **абака** в V в. до н. э. в Древней Греции. Идею такого устройства греки позаимствовали у древних вавилонян. Более поздний римский вариант абака изображен на рисунке 2.21. Абак представлял собой доску с параллельными вертикальными углубленными отрезками, в которые ложились какие-то предметы, чаще всего камни.



Рис. 2.21. Римский абак

Подобные устройства в дальнейшем были созданы в разных странах: Китае **(суаньпань)**, Японии **(соробан)**, России **(счеты** (рис. 2.22)) – и использовались до конца XX в.



Рис. 2.22. Счеты

Вместе с простыми и довольно распространенными устройствами типа абака еще в Древней Греции и Риме применяли различные вычислительные устройства с использованием колесиков с зубцами. Об этом свидетельствует находка с затонувшего античного судна. Ее назвали «Антикитерский механизм» по названию греческого острова, вблизи которого нашли судно в 1902 г. Но только во второй половине XX в. ученые смогли по остаткам механизма восстановить его строение и определить назначение (рис. 2.23). Созданное в 150–100 гг. до н. э. устройство использовалось как календарь, а также для автоматического определения положения небесных тел – Солнца, Луны и известных грекам планет – Меркурия, Венеры, Марса, Юпитера, Сатурна.



Рис. 2.23. Антикитерский механизм: фрагмент находки и компьютерная модель

Однако, к сожалению, устройства, подобные антикитерскому механизму, и научные знания, необходимые для их изготовления, были на длительное время потеряны. Активные работы по совершенствованию вычислительных устройств начались только в XVII в. с развитием промышленности, торговли, строительства, военного дела.

В 1642 г. французский математик, физик, инженер и философ **Блез Паскаль** (1623–1662) подал на рассмотрение королевского совета свое механическое вычислительное устройство. В дальнейшем он создал несколько десятков таких устройств (рис. 2.24). Их использовали для выполнения сложения и вычитания чисел.

Впоследствии идеи Паскаля были развиты и усовершенствованы многими математиками и инженерами. В частности, немецкий ученый **Готфрид Лейбниц** (1646–1716) создал первый в мире арифмометр – механическое вычислительное устройство, которое выполняло все четыре арифметических действия (рис. 2.25).

Усовершенствованные варианты арифмометров различных конструкций активно использовались на протяжении нескольких веков и были заменены электронными калькуляторами во второй половине XX в.



Рис. 2.24. Вычислительное устройство Блеза Паскаля



Рис. 2.25. Арифмометр Готфрида Лейбница

Одновременно с усовершенствованием арифмометров ученые пытались создать вычислительные устройства, которые могли бы автоматически проводить вычисления, выполняя команды заранее разработанной программы. Одним из первых идею создания автоматизированной машины для вычислений выдвинул английский математик и конструктор **Чарльз Беббидж** (1792–1871) (рис. 2.26). Он назвал свое устройство **аналитической машиной**.



Рис. 2.26. Чарльз Беббидж

Согласно проекту Беббиджа, аналитическая машина имела следующие составные части:

1. **«Склад»** для хранения чисел (в современной терминологии – **память**).
2. **«Мельница»** для выполнения арифметических операций над числами (в современной терминологии – **арифметическое устройство**).
3. Устройство, которое управляет последовательностью выполнения арифметических операций (в современной терминологии – **устройство управления**).
4. Устройство для **ввода начальных (входных) данных**.
5. Устройство **вывода результатов**.



Рис. 2.27. Ада Лавлейс

Ада Лавлейс (1815–1852) (рис. 2.27), дочь известного английского поэта Джорджа Байрона, работала вместе с Беббиджем над созданием проекта аналитической машины и его реализацией. Она впервые описала основные принципы разработки программ для вычислительных машин. В связи с этим Ада Лавлейс считается первым в мире программистом, и в ее честь назван один из современных языков программирования **Ada**.

К концу XIX в. устройства для вычислений были ручными или механическими (рис. 2.28). И только в конце XIX в. американский ученый **Герман Холлерит** (1860–1929) предложил новое устройство, работа которого основывалась на ис-

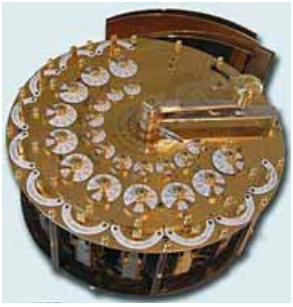


Рис. 2.28. Вычислительная машинка И.Г. Мюллера

пользовании электрического тока, – **табулятор** (рис. 2.29). Он был предназначен для обработки данных переписи населения. Данные о конкретном человеке записывали не на бумаге, а отмечались отверстиями в строго определенных местах персональной карты – перфокарты.



Рис. 2.29. Табулятор Германа Холлерита

Перфорация (лат. *perforate* – продырявливать) – пробивка отверстий, совокупность отверстий.

Основанная Холлеритом в 1896 г. фирма по массовому выпуску табуляторов, претерпев ряд реорганизаций и смену собственников, с 14 февраля 1924 г. стала называться **IBM Corporation**. Сегодня – это один из основных производителей мощных компьютеров.

До середины XX в. разрабатывались различные конструкции механических и электромеханических вычислительных устройств. Они позволяли значительно ускорить процесс вычисления.

В конце 30-х и в 40-х годах XX в. ученые и конструкторы разных стран создали качественно новые вычислительные машины, в которых, как правило, использовались электрические устройства. Так, в Германии в 1941 г. **Конрад Цузе** (1910–1995) (рис. 2.30) создал первую вычислительную машину на электромеханических реле, которую назвал *Z3*. В ней использовалось двоичное кодирование. В 1950 г. он создал компьютер на электронных лампах – *Z4*.



Рис. 2.30. Конрад Цузе

В 1941 г. в США **Джон Винсент Атанасов** (1903–1995) вместе со своим ассистентом **Клиффордом Берри** (1918–1963) создал первый компьютер на электронных лампах с использованием двоичного кодирования – **ABC** (англ. *Atanasoff Berry Computer*). Чуть позже (1939–1944), также в США, **Говард Эйкен** (1900–1973) в сотрудничестве с IBM создал вычислительную машину **Mark-1** на электромагнитных реле с автоматическим управлением последовательностью операций.

В 1943–1944 гг. в условиях полной секретности ученые Великобритании создали вычислительную машину **Colossus**, предназначенную для дешифровки радиограмм фашистской Германии. От других машин того времени ее отличало то, что программа, по которой она работала, сохранялась в памяти самой машины.

В 1943–1946 гг. в США **Джон Моучли** (1907–1980) и **Преспер Эккерт** (1919–1995) создали **ENIAC** (англ. *Electronic Numerical Integrator and Calculator* – электронный цифровой интегратор и вычислитель) – **электронную вычислительную машину** (ЭВМ), которая содержала 18 000 электронных ламп, весила 30 т и выполняла 5000 операций в секунду (рис. 2.31). После завершения этого проекта они сразу начали работу над новым

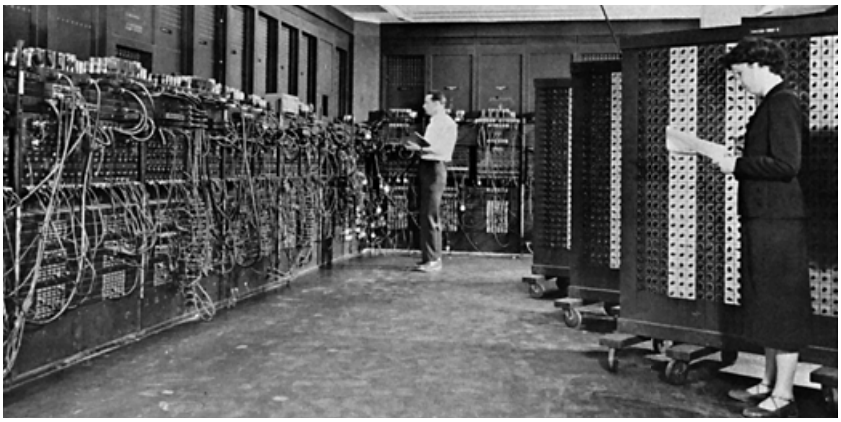


Рис. 2.31. Вычислительная машина **ENIAC**

компьютером по заказу военного ведомства США – **EDVAC** (англ. ***Electronic Discrete Variable Automatic Computer*** – электронный дискретно переменный автоматический вычислитель). К разработке привлекли известного американского математика **Джона фон Неймана** (1903–1957), который в статье «Предварительный доклад о машине EDVAC» сформулировал основные принципы построения универсальной вычислительной машины. Несмотря на то, что эти идеи разработали Эккерт и Моучли, они стали известны всему миру как «принципы фон Неймана».

Широкое распространение компьютеры получили после появления большого количества сравнительно дешевых вычислительных машин – так называемых мини-ЭВМ, которые в дальнейшем стали называться персональными компьютерами. Первые из них продавались в виде электронных конструкторов, из деталей которых пользователь мог самостоятельно собрать компьютер, используя телевизор в качестве монитора. Первыми персональными стали компьютеры, разработанные фирмами: **MITS** (англ. ***Micro Instrumentation and Telemetry Systems*** – системы микроинструментов и телеметрии) – компьютер **Altair 8800** (1975 г.); **Apple Computer Company** – компьютер **Apple** (1976 г.); **IBM** – компьютер **IBM PC** (1981 г.). Все более увеличивающееся распространение персональных компьютеров значительно расширяет круг задач, решаемых с использованием ЭВМ.

Подробнее историю развития вычислительной техники можно рассмотреть в хронологической таблице в приложении 3, которое можно просмотреть на сайте **Інформатика для всіх** (*allinf.at.ua*) в разделе **Матеріали до підручників. 8 клас. Підручник.**



Рис. 2.32. **Altair 8800**



Рис. 2.33. **Apple I**



Рис. 2.34. **IBM PC**

ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ В УКРАИНЕ

В 1951 г. в Киеве под руководством **Сергея Алексеевича Лебедева** (1902–1973) была введена в действие универсальная ЭВМ с программой, хранившейся в памяти, – **МЭСМ** (***Малая Электронная Счетная Машина***). Это была первая ЭВМ в Советском Союзе. Она имела такие характеристики:

- общее количество электронных ламп – около 6000;
- двоичная система кодирования данных;
- количество разрядов – 16 для кодирования числа и один для кодирования знака;
- емкость запоминающего устройства – 31 число и 63 команды;
- быстродействие – около 3000 операций в минуту.

Первые программы для этой ЭВМ составила **К.Л. Ющенко** (1919–2001).

В Киеве разработку вычислительных машин продолжили в Институте кибернетики, который возглавил выдающийся ученый **В.М. Глушков** (1923–1982). Под его руководством завершена разработка ЭВМ **Киев** (1959 г., разработчики Б.В. Гнеденко, Л.М. Дашевский, К.Л. Ющенко), создана серия универсальных ЭВМ **Днепр** (1961 г., главный конструктор Б.Н. Малиновский), ЭВМ для инженерных расчетов **Луч** (1963 г.), серии ЭВМ для инженерных расчетов **МИР** (***машина для инженерных расчетов***, 1960-е гг.) и ряда ЭВМ для военных целей. За выдающиеся достижения в развитии информатики в 1996 г. Международное компьютерное сообщество (IEEE Computer Society) наградило В.М. Глушкова медалью «Компьютерный пионер» (рис. 2.35).

Пионер (англ. *pioneer* – первый поселенец, исследователь, первооткрыватель) – человек, который первым прокладывает дорогу в какой-нибудь новой области деятельности.



Рис. 2.35. Лицевая и оборотная стороны медали В.М. Глушкова «Компьютерный пионер»

Значительный вклад в развитие компьютерной техники для ракетной отрасли сделали коллективы научно-производственных объединений, конструкторских бюро из разных городов Украины. Так, в Северодонецком НПО «Импульс» производились электронные вычислительные машины для систем управления производством М6000–М7000, клавишные ЭВМ **Искра**, средства для автоматического управления баллистическими ракетами. В Киевском НПО «Кристалл» спроектирован, а в г. Светловодске производился первый в Советском Союзе и Европе микрокалькулятор **Электроника** (рис. 2.36).



Рис. 2.36. Микрокалькулятор

В 70–90-е гг. XX в. Харьковское НПО «Хартрон» и Киевский радиозавод разрабатывали и изготавливали ЭВМ для установки на ракетно-космических комплексах.

Разработана конструкция и организован серийный выпуск ЭВМ «Карат» (Киевский научно-исследовательский институт радиоэлектроники и Киевский завод «Буревестник», 70–80-е гг.

XX в.) для управления надводными и подводными кораблями.

В 1975 г. в Институте кибернетики АН УССР под руководством Н.М. Амосова спроектирован первый в СССР автономный транспортный робот ТАИР (рис. 2.37), который мог двигаться в естественной среде, минуя преграды.



Рис. 2.37. Конструкторы налаживают робот ТАИР

Ученые и инженеры Украины в течение 60–80-х гг. XX в. разработали много компьютеров для различных ракетных комплексов. Так, для крупнейшей в мире баллистической ракеты Р-36М2 («Сатана») производства Днепропетровского НПО «Южный машиностроительный завод» специалисты Харьковского НПО «Электроприбор» и ВО «Киевский радиозавод» создали бортовую ЭВМ, которая обеспечивала управление запуском ракеты и ее полетом (рис. 2.38).

Сейчас эту ракету используют для вывода на орбиту Земли спутников по международной программе «Днепр».



Рис. 2.38. Пуск ракеты Р-36М2

ВИДЫ СОВРЕМЕННЫХ КОМПЬЮТЕРОВ И ИХ ПРИМЕНЕНИЕ

Вы знаете, что самыми распространенными сегодня являются так называемые **персональные компьютеры** (ПК), среди которых различают **стационарные** и **мобильные (портативные)**. Стационарные ПК в свою очередь делятся на **офисные**, **домашние**, **игровые** и т. п. А мобильные – на **ноутбуки**, **нетбуки**, **планшетные ПК**, **смартфоны** и т. п.

Для решения задач, требующих больших объемов сложных вычислений, например для расчетов прогноза погоды на несколько дней или недель, используют мощные компьютеры. Крупнейшие из них называют **суперкомпьютерами**. Так, в середине 2015 г. самым мощным компьютером в мире был компьютер **Tianhe-2** (рис. 2.39), созданный в Китайском национальном университете оборонных технологий (Гуанчжоу). Он построен на базе процессоров Intel Xeon с общим количеством ядер 3 млн 120 тыс.

В наше время компьютеры используют в науке, промышленности, торговле, управлении, банковской системе, образовании, медицине, транспорте, связи, сельском хозяйстве, системе социального обеспечения и других отраслях деятельности человека.



Рис. 2.39. Компьютер **Tianhe-2**

Примером применения компьютеров в научной сфере является проведение так называемых компьютерных экспериментов. Ведь проведение многих научных исследований связано со значительными трудностями – материальными, техническими, энергетическими и т. п. Например, чтобы изучить процессы, происходящие во время расщепления атомных ядер, нужно строить атомные реакторы, а этот процесс длительный, сложный, опасный и требует больших затрат денежных средств. Во многих случаях создать соответствующие реальные условия эксперимента вообще невозможно, например, невозможно управлять процессами, которые происходят в какой-то далекой галактике. В таких случаях явление изучают по его **информационной модели** с использованием компьютерной техники и соответствующего программного обеспечения.

Важную роль играет компьютер на производстве. Моделирование и конструирование различных изделий с использованием компьютера значительно сокращает срок их разработки, повышает их эффективность и качество, снижает стоимость. Например, если до применения компьютера в проектировании новой модели автомобиля с момента возникновения идеи до ее производства проходило 5–6 лет, то теперь – менее одного года.

В сфере обслуживания компьютер используют для хранения и обработки различных данных: текстов, таблиц, баз данных, рисунков и фотографий, мультимедийных данных. Трудно назвать такое учреждение, где бы не использовали компьютеры. Картотеки в библиотеках и больницах, выполненные на основе компьютерной базы данных, во много раз надежнее и удобнее в работе, чем традиционные бумажные. Бухгалтер сегодня использует компьютер и за несколько минут имеет результаты, на получение которых раньше ему были нужны часы или дни. Банкир, не выходя из своего кабинета, имеет возможность следить за состоянием дел на бирже, простым нажатием клавиши перевести на любой счет определенную сумму денег.

Основными направлениями использования компьютерной техники являются:

- выполнение объемных вычислений и вычислений с высокой точностью;
- создание компьютерных моделей объектов и проведение компьютерных экспериментов;
- обеспечение функционирования автоматизированных систем управления;
- обеспечение хранения и обработки больших объемов данных;
- обеспечение быстрого обмена данными;
- управление промышленной, бытовой и военной техникой с использованием встроенных компьютеров;
- поддержка изучения учебных предметов;
- организация дистанционной формы обучения учащихся, студентов, специалистов, особенно полезно для удаленных малых населенных пунктов, для людей с ограниченными возможностями и т. п.

КОНФИГУРАЦИЯКОМПЬЮТЕРАВСООТВЕТСТВИИСПОТРЕБНОСТЯМИ

В зависимости от того, для каких целей будет использоваться компьютер, подбирают его составляющие (конфигурацию). При этом особое внимание обращают на значение следующих свойств:

- ***мощность процессора*** – определяется значением нескольких свойств – тактовой частотой, количеством ядер, объемом кэш-памяти второго и третьего уровней. Чем больше значения этих свойств, тем производительнее процессор;
- ***объем оперативной памяти*** – чем больше значение, тем выше производительность компьютера;
- ***емкость накопителя на жестких магнитных дисках*** – в зависимости от объемов данных, с которыми работает пользователь: для видео-, графических и аудиоданных нужны НЖМД большей емкости, чем для работы с текстовыми данными;
- ***наличие отдельного видеоадаптера*** – положительно влияет на скорость обработки видеоданных. А скорость обработки данных видеоадаптером зависит от производительности процессора видеоадаптера и объема видеопамяти;
- ***качество отображения данных монитором***, которое зависит от размеров монитора, разрешения и скорости отклика.

Конечно, могут существенно влиять на выбор компьютера и значения других свойств, таких как наличие устройства для работы с оптическим диском, особенностей материнской платы, продолжительность работы в автономном режиме (для мобильных устройств), наличие средств подключения к компьютерным сетям разного типа и т. п. На выбор компьютеров будет существенно влиять и их стоимость.

Значительная часть интернет-магазинов предлагает отбирать компьютеры, используя так называемые «умные фильтры». Например, сайт **Hotline** (*http://hotline.ua/computer/*) в зависимости от назначения предлагает следующие группы компьютеров: *ПК начального уровня*, *Компьютер для работы\учебы*, *Рабочая станция*, *Базовая оптимальная конфигурация*, *Универсальная оптимальная конфигурация*, *Прогрессивная оптимальная конфигурация*, *Мощный игровой ПК*. Сайт **Розетка** (*http://rozetka.com.ua/computers-notebooks*) предлагает только три группы компьютеров: *Начальный уровень*, *Для работы и учебы*, *Игровые компьютеры*.

Стоимость системных блоков стационарных компьютеров при этом изменяется в довольно широком диапазоне – от *2,5* до *120* тыс. грн. Приведем примеры значений основных свойств настольных персональных компьютеров для предложенной выше классификации (табл. 2.5).

Таблица 2.5

Примеры значений свойств компьютеров различного назначения

Название свойства	***Значения свойств для компьютеров различного назначения***		
	Офисный (для учебы и работы)	*Домашний*	*Игровой*
Тип компьютера	Стационарный	Стационарный	Стационарный
Модель процессора	Intel Celeron Dual-Core E3400	AMD FX-8320	Intel Core i7–4790K
Тактовая частота процессора, *ГГц*	2,6	3,5	4,0

Продолжение таблицы 2.5

Название свойства	***Значения свойств для компьютеров различного назначения***		
	Офисный (для учебы и работы)	*Домашний*	*Игровой*
Объем оперативной памяти, *Гбайт*	2	8	16
Емкость жесткого диска, *Гбайт*	250	1000	1000
Тип видеоадаптера	Интегрированный Intel HD Graphics	AMD Radeon R9 270X	NVIDIA GeForce GTX 980 Ti
Объем видеопамяти, *Гбайт*	Дополнительная отсутствует	2	6
Тип звукового адаптера	Интегрированный	Creative X-Fi Xtreme Audio PCI Express	Интегрированный HD Audio 7.1
Монитор	LG 19M45A	Philips 233V5LSB	ASUS VX238H
Диагональ монитора, *дюймов*	18,5	23	23
Разрешение	1366 × 768	1920 × 1080	1920 × 1080
Время отклика, *мс*	5	4	1

Самое важное в этом пункте

Первые устройства для проведения вычислений были созданы человеком несколько тысячелетий назад. Значительный вклад в развитие вычислительной техники сделали Б. Паскаль, Г. Лейбниц, Ч. Беббидж, А. Лавлейс, Г. Холлерит. Первые электронные вычислительные машины были созданы в 40–50-х гг. XX в. К. Цузе (**Z4**), Д.В. Атанасовым и К. Берри (**ABC**), Г. Эйкеном (**Mark-1**), группой английских ученых (**Colossus**), Д. Моучли и П. Эккертом (**ENIAC**), коллективом советских ученых под руководством С. Лебедева (**МЭСМ**).

В Украине разработали первые в Советском Союзе компьютеры. В разработке ЭВМ для различных отраслей экономики и для управления военной техникой участвовали десятки предприятий и организаций из разных уголков Украины. Ведущей организацией стал Институт кибернетики Академии наук Украины под руководством В.М. Глушкова.

Среди компьютеров, в зависимости от назначения и скорости обработки, выделяют **суперкомпьютеры** и **персональные компьютеры**. Последние делят на **стационарные** и **мобильные**. Стационарные компьютеры делят на **офисные**, **домашние**, **игровые** и т. п. А мобильные – на **ноутбуки**, **нетбуки**, **планшетные ПК**, **смартфоны** и т. п.

Дайте ответы на вопросы

1°. Назовите этапы развития информационных технологий.
2•. Опишите, как менялись средства обработки сообщений на разных этапах развития информационных технологий.
3•. Объясните, в чем заслуга Б. Паскаля в развитии устройств для проведения вычислений.
4°. Чем отличался арифмометр Лейбница от устройства Паскаля?
5•. Опишите составляющие универсальной машины для вычислений, предложенной Ч. Беббиджем.
6°. Кого считают первым программистом в мире? Почему?
7•. Опишите период создания первых электронных вычислительных машин (40–50-е гг. XX в.). Какие ученые внесли значительный вклад в развитие электронных вычислительных машин в этот период?
8°. Назовите первые персональные компьютеры. Когда их создали?
9•. Опишите развитие компьютерной техники в Украине.
10*. Объясните, почему существует большая разница в значениях свойств различных компьютеров. Приведите примеры классификации персональных компьютеров.
11•. Что влияет на выбор компьютера для определенных сфер применения?

Выполните задания

1•. Подготовьте схему классификации компьютеров, используя любую прикладную программу, которую вы знаете.
2•. Выполните в Интернете или в печатных изданиях поиск сведений о первых электронных вычислительных машинах и заполните сравнительную таблицу:

Свойства	*Значения свойств ЭВМ*		
	Z4	Colossus Mark2	ABC
Разработчики			
Количество электронных ламп			
Использование двоичного кодирования			
Хранение программы в памяти			

3*. Сравните значения основных свойств первых электронных вычислительных машин: американской ENIAC и созданной в Украине МЭСМ. Результаты сравнения опишите в виде таблицы в тетради.
4*. Подготовьте сообщение о разработке украинскими учеными ЭВМ для управления атомными ледоколами.
5•. Подберите компьютер для использования в качестве домашнего мультимедийного центра. Свой выбор обоснуйте.

ПРАКТИЧЕСКАЯ РАБОТА № 2
«Конфигурация компьютера в соответствии с потребностями»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Подберите компьютеры для людей разных профессий и разных сфер применения:
   - писатель, работая в своем кабинете, использует компьютер для создания своих прозаических произведений, поиска сведений, для отправки и получения электронных писем и т. п.;
   - студент использует компьютер для подготовки учебных заданий, а также для создания собственной фонотеки, создания и редактирования учебных видеофильмов;
   - корреспондент интернет-издания использует компьютер для написания и своевременной отправки в редакцию сообщений с новостями во время командировок в разные уголки мира.
2. По результатам выполнения задания заполните таблицу значений свойств.

Название свойства	***Значения свойств для компьютеров***		
	писателя	*студента*	*корреспондента*
Тип компьютера			
Модель процессора			
Тактовая частота процессора, *ГГц*			
Объем оперативной памяти, *Гбайт*			
Емкость жесткого диска, *Гбайт*			
Тип видеоадаптера			
Объем видеопамяти, *Гбайт*			
Тип звукового адаптера			
Монитор			
Диагональ монитора, *дюймов*			
Разрешение			
Время отклика, *мс*			

3. В текстовом процессоре создайте файл, в котором обоснуйте ваш выбор конфигурации для этих компьютеров.
4. Файл с обоснованием отправьте по электронной почте учителю.

2.4. КЛАССИФИКАЦИЯ И ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1. Какие компьютерные программы вы уже знаете? Для чего они предназначены?
2. Что такое операционная система? Для чего она предназначена?
3. Опишите способы запуска программ.

КЛАССИФИКАЦИЯ И ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Вы уже знаете, что существуют различные программы для компьютеров. Есть программы для обработки текстовых, графических, мультимедийных данных, есть программы для управления работой компьютера, его отдельных устройств, таких как принтер, сканер, монитор и др. Совокупность программ для компьютеров еще называют **программным обеспечением (ПО)**. Только под управлением программного обеспечения компьютер способен обрабатывать различные данные.

Современное программное обеспечение делят на следующие виды (рис. 2.40).

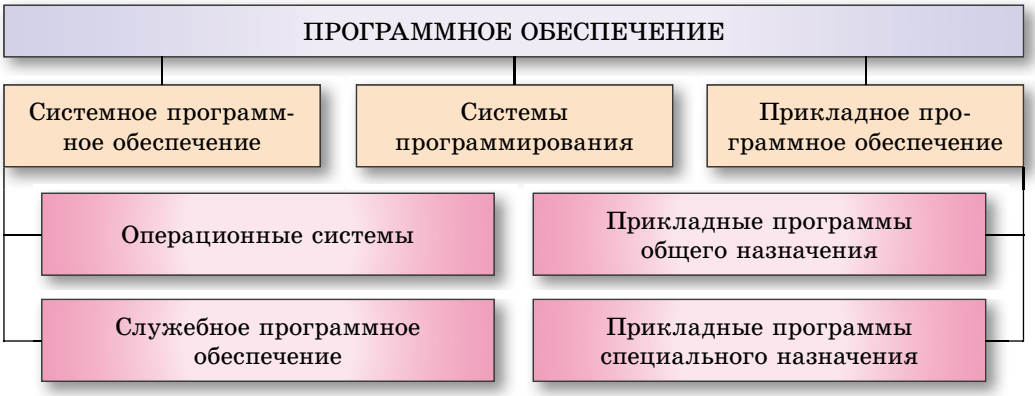


Рис. 2.40. Классификация программного обеспечения

Системное программное обеспечение предназначено для управления работой устройств компьютера и обменом данными между ними, диагностики и устранения неполадок в работе компьютера, автоматизации процесса обработки данных, организации обмена данными между пользователем и компьютером. В состав системного программного обеспечения входят операционные системы и различные утилиты.

Системы программирования предназначены для создания новых программ с использованием различных языков программирования, например **Scratch 1.4, Free Pascal 2.6, DEV-C++ 5.11, Microsoft Visual Studio 2013 Professional, Android Studio 1.4.0, Lazarus 1.4.4, Python 2.6.1** и др.

Прикладное программное обеспечение предназначено для реализации конкретных задач обработки данных, которые пользователь решает в ходе

своей деятельности. Их делят на прикладные программы общего и специального назначения.

К прикладным программам **общего назначения** относятся программы, которые могут применяться в различных отраслях человеческой деятельности для обработки текстов, рисунков, мультимедийных данных, электронных таблиц, создания презентаций и т. п.

Прикладные программы **специального назначения** используются для реализации задач обработки данных в определенной области деятельности, на конкретном предприятии, в организации, фирме или их подразделении. К такому типу программ относятся программы для создания видеоэффектов при производстве кинофильмов, чертежей машин и механизмов в конструкторских и проектных бюро, диагностирования заболеваний в медицинских учреждениях, создания школьного расписания уроков и т. п.

ЛИЦЕНЗИИ НА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ИХ ТИПЫ

На компьютерные программы распространяется действие Закона Украины «Об авторском праве и смежных правах», и их использование возможно только при условии соблюдения требований этого закона, а также требований лицензии, с которой пользователь соглашается, устанавливая программу на свой компьютер.

Есть несколько видов лицензий на компьютерные программы:

- ***коммерческая лицензия*** – предполагает, что пользователь оплачивает стоимость использования данной программы на одном или нескольких указанных в лицензии компьютерах, среди них различают:
  - *«коробочные» версии* лицензии продаются, как правило, для одного экземпляра программы, содержат носитель, на котором записана программа, и инструкцию к ее использованию вместе с ключом для установки (рис. 2.41);
  - *OEM* (англ. ***O**riginal **E**quipment **M**anufacturer* – оригинальный производитель оборудования) – лицензия, предоставляемая на один экземпляр программы вместе с определенным компьютерным оборудованием, например с ноутбуком; подтверждением лицензии является специальная наклейка (рис. 2.42);
  - *корпоративная лицензия* – лицензия на несколько копий программы для использования, например на компьютерах учебных заведений;
- ***пробная,*** или ***trial*** (англ. *trial* – испытание), – лицензия на коммерческое ПО, которая предоставляется для пробного ис-

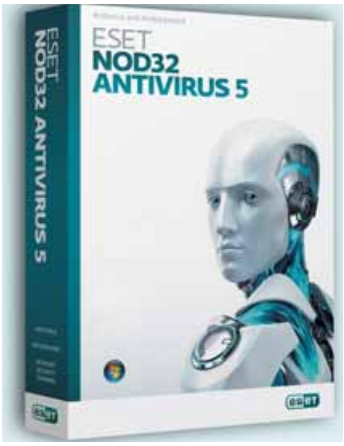


Рис. 2.41. «Коробочная» версия антивирусной программы



Рис. 2.42. Наклейка OEM-версии лицензии операционной системы

пользования программы в течение определенного времени (обычно около месяца) или на определенное количество запусков программ. При этом могут работать не все функции программы;

- ***свободного использования***, или ***freeware*** (англ. *freeware* – свободный товар), – лицензия предусматривает свободное использование программ без выплат вознаграждения автору, но не предусматривает возможности внесения изменений в программу;
- ***с открытым кодом***, или ***free*** (англ. *free* – свободный) ***software***, или ***libre*** (исп. *libre* – свободный) ***software***, – лицензия на свободное программное обеспечение, предусматривающее не только бесплатное использование программ, но и право на их модификацию, внесение изменений в программу.

КЛАССИФИКАЦИЯ И ОСНОВНЫЕ ФУНКЦИИ ОПЕРАЦИОННЫХ СИСТЕМ

Как вы уже знаете из курса информатики шестого класса, существует много различных операционных систем (ОС). В зависимости от назначения их подразделяют на ***операционные системы для стационарных персональных компьютеров*** (**Windows 10**, **OS X** и т. п.), ***для мобильных компьютеров*** (**Windows Mobile 6.0**, **iOS 9.0**, **Android 6.0** и т. п.), ***для серверов и суперкомпьютеров*** (**Unix**, **Linux**, **Windows Server 2012**, **Solaris 11** и т. п.).

Также операционные системы можно разделить по типу лицензий, на основании которых они распространяются. Так, операционные системы **Windows** имеют коммерческую лицензию и свободно не распространяются, однако некоторые из них могут использовать *trial*-лицензию. К ОС, распространяющимся по лицензии ***free software*** – свободного программного обеспечения, в первую очередь относятся различные версии операционной системы **Linux**.

Операционные системы предназначены для управления работой устройств компьютера и обеспечения обмена данными между ними, хранения данных в оперативной памяти и на внешних носителях, обеспечения выполнения других программ и обмена данными между пользователем и компьютером.

ПОНЯТИЕ ИНСТАЛЛЯЦИИ И ДЕИНСТАЛЛЯЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для установки ОС на компьютере нужно выполнить специальную операцию, которую называют **инсталляцией операционной системы**.

Для инсталляции ОС необходимо иметь оптический диск или флэш-носитель с установочным пакетом – **дистрибутивом** (англ. *distribute* – распределять). При загрузке с такого диска или флэш-носителя программа установки ОС спросит у пользователя разрешение на установку и на следующих шагах предложит ввести ряд дополнительных данных – диск, на который будет установлена ОС, язык и набор региональных стандартов и т. п. Например, для начала процесса инсталляции ОС **Linux Ubuntu** нужно выбрать пиктограмму **Установить Ubuntu** на **Рабочем столе** или на **Панели запуска** (рис. 2.43). Существуют и другие варианты выполнения инсталляции, например с использованием локальной сети или Интернета.



Рис. 2.43. Рабочий стол **Linux Ubuntu** с ярлыками программы установки

При установке происходит размещение составных ОС на выбранном диске, налаживание ее взаимодействия с аппаратной составляющей компьютера. Диск, на который установлена операционная система, называют **системным**.

Любое программное обеспечение, используемое в современных компьютерах, работает под управлением операционной системы. Каждая из программ создается для определенной операционной системы и не может работать с другой, не совместимой. Перед началом использования компьютерной программой предстоит процесс согласования с операционной системой. Этот процесс также, называемый **инсталляцией**, происходит в соответствии с алгоритмом, записанным в специальной программе, например, ***setup.exe, install.exe*** (для **Windows**) или ***INSTALL.sh***, а также файлов с расширением ***deb*** (для **Linux**).

В дальнейшем необходимо следовать указаниям программы установки. Если инсталляция прошла успешно, то команда для запуска программы будет включена:

- для **Windows** – в меню **Пуск** (как правило, в раздел **Все приложения**) и в список уже установленных программ;

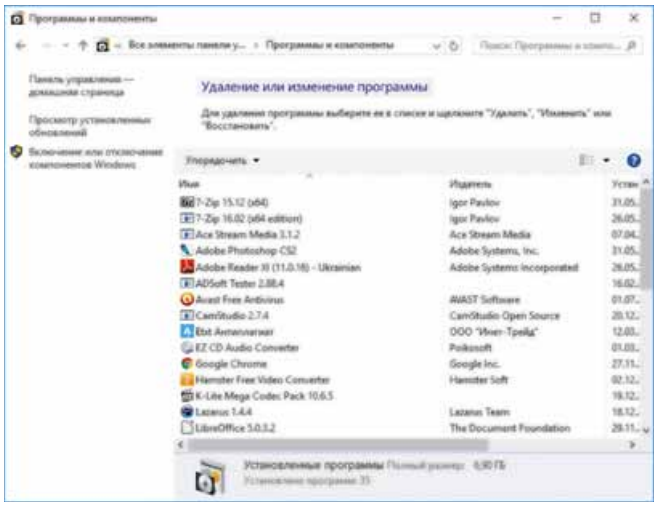


Рис. 2.44. Окно **Программы и компоненты**

- для **Linux Ubuntu** – в **Панель запуска** и в **Центр приложений Ubuntu**.

Удаление **(деинсталляция)** программ проводится:

- для **Windows** – в окне **Программы и компоненты** (***Пуск*** ⇒ ***Все приложения*** ⇒ ***Служебные – Windows*** ⇒ ***Панель управления*** ⇒ ***Программы и компоненты*** для **Windows 10** или ***Пуск*** ⇒ ***Панель управления*** ⇒ ***Программы и компоненты*** для **Windows 7**) следует выбрать программу,

которую нужно удалить или изменить ее настройки, после чего – кнопку **Изменить/Удалить** (рис. 2.44);

- для **Linux Ubuntu** – в окне **Центра приложений Ubuntu** (***Панель запуска*** ⇒ ***Центр приложений Ubuntu***) выбрать программу, которую необходимо удалить, и выбрать кнопку **Удалить** (рис. 2.45).

Выбор операции удаления запускает программу, которая корректно удаляет все компоненты программы с компьютера.

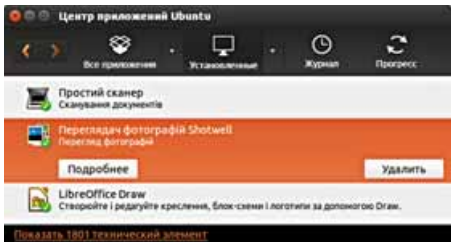


Рис. 2.45. Фрагмент окна **Центра приложений Ubuntu**

ПРОБЛЕМЫ СОВМЕСТИМОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Компьютерные программы разрабатываются для определенной операционной системы, в соответствии со стандартами, которые устанавливают разработчики ОС. Программы, разработанные для одного типа операционных систем, как правило, несовместимы с другими типами или версиями операционных систем и не могут быть установлены. Так, программы, разработанные для **Windows**, не могут работать в ОС семейства **Linux**. Для того чтобы определенная программа заработала в другой операционной системе, ее код нужно изменить с учетом требований другой операционной системы. Так, существуют, например, браузеры **Chrome**, для **Windows**, **Linux**, **Android** и т. п. (рис. 2.46).

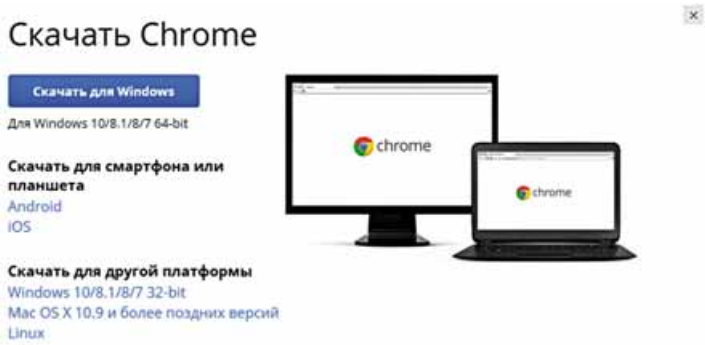


Рис. 2.46. Фрагмент веб-страницы загрузки браузера **Google Chrome**

Определенное программное обеспечение не всегда корректно работает даже под управлением операционных систем одного семейства, но разных его версий. Так, отдельные программы для **Windows 10** не смогут работать с **Windows XP**.

СЛУЖЕБНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Служебное программное обеспечение – это программы, предназначенные для диагностирования аппаратной и программной составляющих компьютера, расширения возможностей ОС. При необходимости они устра-

няют недостатки и оптимизируют работу компьютера. Эти программы называют **утилитами** (англ. *utility* – полезность). Часть таких программ включается в состав операционных систем при инсталляции ОС. Например, в состав операционной системы **Windows 10** входят такие утилиты, как **Диспетчер задач, Восстановление системы, Оптимизация дисков, Очистка диска, Монитор ресурсов, Сведения о системе, Планировщик заданий, Панель управления** и др.

Аналогичные программы есть и в других операционных системах. Так, в **Linux Ubuntu** такими программами, например, являются: **Менеджер архивов, Журнал системы, Анализатор использования диска, Резервное копирование, Системный монитор, System Testing, Диски** и др.

Дополнительные утилиты можно приобрести или свободно скопировать с сайтов разработчиков и установить для дальнейшего использования.

ФОРМАТИРОВАНИЕ НОСИТЕЛЕЙ ДАННЫХ

Среди служебных программ есть программы, которые выполняют обслуживание устройств внешней памяти. Их еще называют программами для работы с дисками. К таким программам относятся программы разметки, форматирования, проверки, дефрагментации, очистки носителей данных и др.

Использование таких программ связано с необходимостью обеспечения длительного надежного хранения данных, а также быстрого и точного поиска нужных данных.

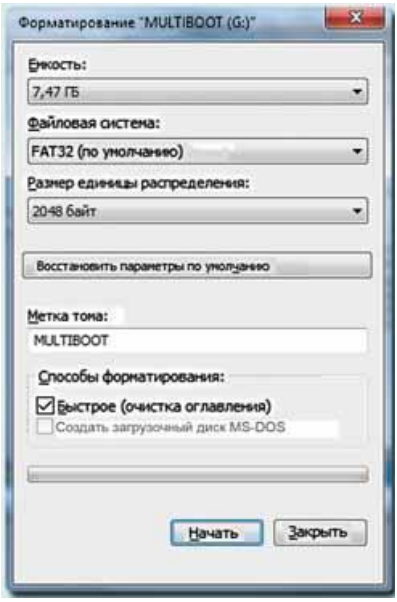


Рис. 2.47. Окно **Форматирование**

Вы знаете, что для того, чтобы быстро что-то найти в хранилище данных, нужно эти данные упорядочено хранить. Для обеспечения упорядоченного хранения данных на жестких магнитных дисках перед использованием его разбивают на части – **логические диски**. Количество логических дисков задает пользователь в зависимости от емкости диска и его назначения. Как правило, это 1–3 логических диска, которые в дальнейшем будут рассматриваться операционной системой как отдельные объекты. Первый из них можно отвести для установки операционной системы, а другие – для хранения данных.

После создания логических дисков нужно выполнить операцию **форматирования** каждого из них. В ходе форматирования выполняется разбиение диска на отдельные «ячейки памяти». Такие «ячейки» называют **«сектора»**. В каждом секторе может храниться от 0 до 512 байт данных. В ходе форматирования для упрощения доступа к данным на носителях большой емкости секторы объединяют в **кластеры**. Кластер может состоять из одного или нескольких, напри-

мер четырех, секторов. В последнем случае емкость кластера будет: 512 байт · 4 = 2048 байт.

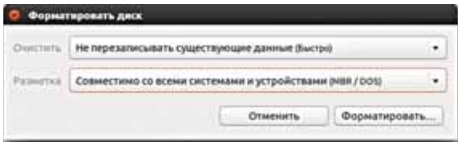
При форматировании также создается структура таблицы размещения файлов на носителе.

Форматирование проводят также и для других внешних запоминающих устройств. Так, для форматирования флэш-носителя нужно выполнить следующую последовательность действий:

1. Открыть окно **Проводника** с перечнем устройств внешней памяти, выполнив – ***Пуск*** ⇒ ***Проводник***.
2. Открыть контекстное меню устройства флэш-памяти.
3. Выбрать команду **Форматировать**.
4. В открывшемся окне (рис. 2.47) установить:
   - тип файловой системы – в списке **Файловая система** (для использования на компьютерах с разными файловыми системами рекомендуется оставить по умолчанию файловую систему FAT32);
   - емкость кластера – среди предложенных значений в списке **Размер единицы распределения**;
   - имя устройства, которое будет отображаться при просмотре внешних запоминающих устройств, – в поле **Метка тома**;
   - способ форматирования – установлением или снятием отметки флажка **Быстрое (очистка оглавления)**;
5. Выбрать кнопку **Начать**.

Для тех, кто работает с Linux

Форматирование носителей данных в **Linux Ubuntu** можно выполнить по следующему алгоритму:



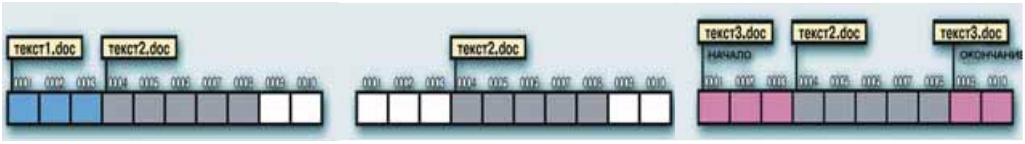
1. Открыть окно **Диски** (***Поиск на компьютере в Интернете*** ⇒ ***Диски***).
2. Выбрать нужный носитель в левой части окна.
3. Выбрать кнопку **Дополнительные действия** ⚙.
4. Выбрать команду **Форматировать**.
5. Установить параметры форматирования в окне **Форматировать диск** (см. рис.):
   - способ форматирования – в списке **Очистить**;
   - тип файловой системы – в списке **Разметка**.
6. Выбрать команду **Форматировать**.
7. Подтвердить выполнение команды форматирования выбором соответствующей кнопки в диалоговом окне.

Для тех, кто хочет знать больше

Современные ОС автоматически выполняют проверку состояния устройств внешней памяти. Особенно это касается накопителя на жестких магнитных дисках – основного устройства хранения данных в персональных компьютерах. Часто возникает потребность в проверке логической структуры и физического состояния носителей данных. Для этого пользователь может воспользоваться стан-

дартной служебной программой проверки носителей, выбрав в контекстном меню устройства хранения данных команду **Свойства** и на вкладке **Сервис** – кнопку **Проверить**.

Чтобы выполнить тщательную проверку диска в ОС **Windows7**, установите флажок **Проверять и восстанавливать поврежденные секторы**. При сканировании выполняется поиск и исправление физических ошибок непосредственно на диске, поэтому этот процесс может длиться гораздо дольше.



а) б) в)

Рис. 2.48. Последовательность операций записи и удаления файлов

Во время записи данных на носитель ОС разбивает файлы на части, соответствующие емкости кластеров, и записывает в первые свободные кластеры на носителе. Если выполняются только операции записи, то файлы будут записываться в соседние кластеры подряд. Однако, если, кроме записи данных, выполняется и удаление файлов, то, как правило, возникает явление ***фрагментации файлов***. Например (рис. 2.48), сначала был записан на диск файл **текст1.doc**, занявший три кластера, затем был записан еще один файл **текст2.doc**, который занял пять следующих кластеров (рис. 2.48, *а*). Впоследствии первый файл был удален и были освобождены три первых кластера (рис. 2.48, *б*). Далее был записан файл **текст3.doc**, который имел размер пяти кластеров. Для записи этот файл был разбит на две части (фрагменты). Первая часть размером в три кластера была записана в кластеры, освободившиеся после удаления первого файла, а вторая (два кластера) – записана в свободные кластеры после файла **текст2.doc** (рис. 2.48, *в*). Чем чаще выполняются операции записи и удаления файлов, тем больше будет фрагментов файлов, которые находятся не в соседних кластерах.

Для устранения такой ситуации и ускорения обмена данными с носителями данных нужно периодически выполнять **дефрагментацию носителей**. В ходе дефрагментации специальная программа пытается разместить каждый из файлов в соседних кластерах диска. Для запуска этой программы можно выбрать кнопку **Оптимизировать** на вкладке **Сервис** окна свойств устройства хранения данных. В открывшемся окне следует выбрать нужный носитель данных и выбрать кнопку **Оптимизировать диск**.

Самое важное в этом пункте

Совокупность программ для компьютеров называют **программным обеспечением (ПО).** Его можно разделить на следующие виды: *системное программное обеспечение*, *системы программирования*, *прикладное программное обеспечение*.

Есть несколько видов лицензий на компьютерные программы: *коммерческая лицензия («коробочная», OEM, корпоративная), пробная (trial), свободного использования (freeware), с открытым кодом (free software).*

В зависимости от назначения операционные системы делят на *операционные системы для стационарных персональных компьютеров*, *для мобильных компьютеров*, *для серверов и суперкомпьютеров*. Также ОС раз-

деляют зависимо *от типа лицензии, в соответствии с которой их распространяют*.

Для установки приложений (включая операционные системы) нужно выполнить специальную операцию, которую называют **инсталляцией**. Программы, разработанные для одного типа операционных систем, как правило, несовместимы с другими типами операционных систем и не могут работать под их управлением.

Служебное программное обеспечение – это программы, предназначенные для диагностирования аппаратной и программной составляющих компьютера, расширения возможностей ОС.

Форматирование носителя данных – это процесс его подготовки к записи данных, заключающийся в создании мест для хранения данных – секторов емкостью 512 байт, которые объединяются в кластеры. В ходе форматирования также создается структура таблицы файлов, в которую будут в дальнейшем заноситься данные о файлах и кластерах, в которых они хранятся.

Дайте ответы на вопросы

1°. Что такое программное обеспечение?
2•. Опишите классификацию программного обеспечения.
3°. Что такое системное программное обеспечение? Приведите примеры программ, входящих в его состав.
4•. Какие программы относятся к системам программирования? Приведите примеры.
5•. Какие из изученных ранее программ относятся к прикладным программам? Какие из них являются прикладными программами общего назначения?
6•. Для каких целей используются прикладные программы специального назначения? Приведите примеры таких программ.
7•. Назовите виды лицензий для компьютерных программ. Чем они отличаются?
8•. Опишите последовательность подготовки носителей данных к использованию на примере жестких магнитных дисков.
9•. Что такое кластер носителя данных? Какие значения может принимать емкость кластера?
10•. Опишите последовательность операций форматирования носителя данных.

Выполните задания

1•. В одной из прикладных программ общего назначения подготовьте схему классификации видов лицензий программного обеспечения.
2•. Емкость одного кластера диска 1024 байт. На диск записали файлы размером 2750 байт и 324 Кбайт. Сколько кластеров займут эти файлы?
3•. Откройте окно программы **Очистка диска (*Пуск* ⇒ *Все приложения* ⇒ *Средства администрирования Windows* ⇒ *Очистка диска*)**. Определите, в чем заключается операция очистки дисков, в каких

группах объектов происходит поиск файлов, которые могут быть лишними. Какой общий размер файлов, от которых программа предлагает очистить диски?

4•. Проведите форматирование одного из сменных носителей (по указанию учителя). При форматировании следуйте приведенному в учебнике алгоритму и установите следующие значения свойств форматирования:

- тип файловой системы – *оставить предложенную операционной системой;*
- емкость кластера – *512 байт;*
- метка тома – *флэшка_Фамилия*;
- способ форматирования – *быстрое*.

5*. Исследуйте различия в реальных размерах файлов и размерах, которые они занимают на носителе данных для трех файлов, указанных учителем, например, расположенных в папке **Глава 2\Задание 2.5**. Заполните таблицу и определите емкость кластера диска.

Имя файла	*Размер файла*	*Размер файла на диске*	*Емкость кластера*

6*. Составьте схему классификации операционных систем по приведенным в учебнике сведениям. Используя материалы Интернета, дополните эту схему классификацией по значению еще одного свойства операционных систем.

2.5. СЖАТИЕ И АРХИВАЦИЯ ДАННЫХ

1. С какими компьютерными программами вы уже умеете работать? Для чего они предназначены?
2. Что такое служебное программное обеспечение? Для чего оно предназначено?
3. Что такое архив? Для чего он используется?

СЖАТИЕ ДАННЫХ, ВИДЫ СЖАТИЯ ДАННЫХ

Часто возникает потребность в уменьшении размеров данных, хранящихся в памяти компьютера. Для этого используют специальные способы сжатия данных, которые называют **алгоритмы (методы) сжатия данных**. Сжатие данных используется при создании файлов определенных типов, например, графических типа **TIFF, JPEG, PNG** или звуковых типа **MPEG3, WMA**, для передачи файлов по сети и т. п.

Сжатие данных **– это процесс перекодировки данных с целью уменьшения размеров файлов.**

Различают алгоритмы сжатия, обеспечивающие сжатие ***без потери данных***, и алгоритмы, предусматривающие ***частичную потерю данных***.

Алгоритмы с частичной потерей данных (другое название «***с регулируемой потерей данных***») используют, когда целостность данных не очень существенна. Например, при сжатии графических, видео-, звуковых файлов, поскольку органы чувств человека не всегда способны заметить незначительное отличие в оттенках цветов на фотографии, в воспроизведении звуковых или видеоданных и т. п. Или потеря в качестве не так важна, как необходимость уменьшить размеры файлов для записи на оптический диск или для использования в Интернете. Однако эти алгоритмы нельзя применять при сжатии текстовых и числовых данных.

Для тех, кто хочет знать больше

Методы сжатия данных

Алгоритмы Д. Хаффмана и ***Шеннона-Фано*** предусматривают перекодирование данных. Символы, которые чаще встречаются в сообщении, перекодируются меньшим количеством символов, а какие реже – большим. Так, например, для кодирования в **Windows-1251** предыдущего предложения, которое содержит 115 символов, включая пробелы, необходимо использовать 1 байт данных на каждый символ, или 115 байт. Если же посчитать, сколько раз тот или иной символ встречается в предложении, то окажется, что всего символов в предложении 28, чаще встречается символ «пробел» – 14 раз, символы «е» и «о» – по 11, «и» – 10, «с» – 7, «в» – 6 и т. д. Меньше всего, один раз, встречаются символы «у», «п», «ж», «–», «.». Если применить алгоритм сжатия данных для двоичного кода (0 и 1) и обозначить символы, чаще встречающиеся одним символом (1 бит): символ «пробел» – 0, а символ «е» – 1, следующие по частоте повторяемости – двумя символами (2 бит): символ «о» – 00, символ «и» – 01 и т. д. Для символов, встречающихся реже всего, будут использованы четыре двоичные цифры или 4 бит. В результате общая длина кода закодированного таким способом предложения будет составлять 90 бит, или 11 байт и 2 бит. Как видим, общий объем кода предложения уменьшится в 10 раз.

Конечно, это довольно упрощенная схема кодирования, не следует забывать, что в сжатом таким способом файле должна содержаться таблица кодирования и дополнительные служебные сведения. Поэтому при кодировании очень маленьких файлов может произойти не уменьшение, а, наоборот, увеличение размеров файла. Для файлов с большим объемом текстовых или цифровых данных этот алгоритм дает высокую степень сжатия.

Алгоритм RLE (англ. ***R**un-**L**ength **e**ncoding* – кодирование повторов или длин серий) используется чаще для кодирования графических данных. Этот метод предусматривает замену последовательности однотипных данных ее сокращенным описанием. Например, в рисунке логотипа корпорации **Apple** (рис. 2.49) использованы только два цвета – черный и белый. Если обозначить черный цвет цифрой «1», а белый – «0», тогда код первой строки изображения должен состоять из 48 цифр «0» (48 бит). Но их можно заменить описанием, указав код символа и количество повторов – 0 48. На код описания будет отведено 9 бит: 1 бит на цифру «0» и по четыре бита на две цифры количества повторов. Второй ряд точек будет закодирован так же, а третий: 0 29 1 4 0 15 – 23 бит. Для цветных изображений с большим количеством одноцветных фрагментов данный алгоритм обеспечит высокую эффективность сжатия.

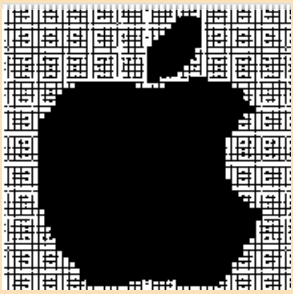


Рис. 2.49. Логотип Apple

АРХИВАЦИЯ ДАННЫХ

Несмотря на повышение надежности компьютеров и носителей данных, полной гарантии сохранения данных они все же не дают. Потеря данных может привести к очень серьезным последствиям. Так, удаление или повреждение данных о вкладах и перечислениях средств клиентов может привести к краху банка, потеря данных о продаже билетов усложнит перевозки пассажиров, уничтожение результатов опытов может свести на нет многолетние научные исследования. Даже потеря записной книжки с номерами телефонов друзей доставит вам хлопот. Поэтому возникает потребность в создании копий данных. Самые важные данные дублируют, записывая на другие жесткие диски, на оптические диски и т. п. Обычно, для удобства использования и уменьшения объемов данных, файлы и папки при создании резервных копий упаковывают в один файл. Такие копии данных называют **архивами**, а файлы, в которые они упаковываются, – **архивными файлами**, или для упрощения – **архивами**.

Создание копий данных с использованием специальных программ, которые могут использовать сжатие данных, называют *архивацией*.

Большинство операционных систем имеют в своем составе средства создания резервных копий данных. Эти программы, как правило, осуществляют копирование данных для восстановления операционной системы на том же или на другом диске, а также создают архивы файлов пользователя.

Для создания резервных копий файлов следует:

1. Открыть окно настроек архивации и восстановления файлов (***Пуск*** ⇒ ***Все приложения*** ⇒ ***Служебные Windows*** ⇒ ***Панель управления*** ⇒ ***Резервное копирование и восстановление***).
2. Выбрать ссылку **Настройка резервного копирования** (для первого запуска резервного копирования или ссылку **Изменить параметры** для последующих).
3. Указать устройство, на которое будет записан архивный файл.
4. Указать перечень папок с файлами, которые будут включены в резервную копию (рис. 2.50).



Рис. 2.50. Фрагмент окна **Настройка архивации**

5. Изменить, при необходимости, расписание осуществления автоматического резервного копирования.
6. Начать процесс создания архива данных пользователя выбором кнопки **Сохранить параметры и запустить архивацию.**

В последних версиях операционных систем появилась возможность создавать **образы системы**, в которые включаются как копии файлов пользователя, так и все настройки операционной системы с установленными ранее программами. Восстановление данных с такого образа обеспечивает быстрое возвращение к прежнему состоянию всей системы без необходимости переустановки программ.

Для создания образа системы в **Windows 10** следует:

1. Выполнить ***Пуск*** ⇒ ***Все приложения*** ⇒ ***Служебные Windows*** ⇒ ***Панель управления*** ⇒ ***Резервное копирование и восстановление.***
2. Выбрать ссылку **Создание образа системы** в левой части окна.
3. Выбрать место сохранения образа и носители (кроме системного, который выбирается обязательно), данные из которых будут включены в образ системы.
4. Выбрать кнопку **Архивировать.**

В результате на выбранном носителе будет создан файл образа системы вашего компьютера с расширением имени **iso.** Если настроено автоматическое выполнение резервного копирования, то по указанному пользователем расписанию будет происходить ***обновление содержания архивных файлов.*** К архивам будут добавляться новые файлы и папки из установленных пользователем мест, а также обновленные версии ранее включенных в архивы файлов.

Для восстановления данных из резервной копии или с образа диска следует в окне ***Резервное копирование или восстановление файлов*** выбрать ссылку, которая запускает процесс восстановления данных **Восстановить мои файлы**, **Восстановить файлы всех пользователей** или **Выбрать другой архив...** В первом случае будут восстановлены данные только пользователя, сеанс которого сейчас идет, во втором – данные всех пользователей компьютера, которые были сохранены, и в третьем – следует указать место размещения файла сохраненной резервной копии данных.

Для тех, кто работает с Linux

Для создания резервных копий следует открыть окно системных параметров (***Панель запуска*** ⇒ ***Параметры системы***) и в группе **Система** запустить программу **Резервные копии** (рис. 2.51). В окне программы можно установить папки, которые будут включены в резервные копии, уточнить

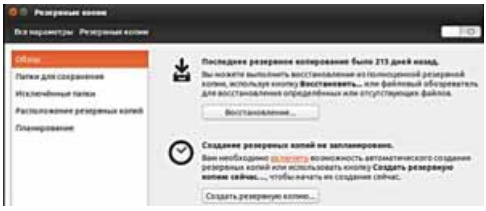


Рис. 2.51. Окно программы **Резервные копии**

размещение хранилища для резервных копий, настроить параметры автоматического резервирования и т. п. Выбрав кнопку **Восстановление**, можно начать восстановление данных с сохраненной ранее резервной копии данных.

Для создания образов системы в **Linux Ubuntu** можно использовать программы работы с оптическими дисками – **Brasero** или **K3b**. Эти программы по умолчанию доступны после установки операционной системы, или их можно загрузить с сайта программного обеспечения **Linux**, используя **Центр программного обеспечения Ubuntu**.

АРХИВАТОРЫ И ТИПЫ АРХИВОВ

Кроме средств операционной системы, существуют другие утилиты, обеспечивающие архивирование данных.

Программы, используемые для выполнения операций над архивами, называют *архиваторами*.

Примерами архиваторов являются программы **WinZIP**, **WinRAR**, **7-zip**, **PowerArchiver**, **WinAce**, **Ark**, **Btar**, **AndroZip**, **FreeArc** и др. Результатом работы этих программ является **архивный файл**, содержащий в сжатом либо не в сжатом виде файлы и папки. В процессе архивации могут быть использованы дополнительные меры по защите данных от несанкционированного доступа, например установка пароля на доступ к данным в архиве.

В зависимости от алгоритмов, в соответствии с которыми выполняется сжатие и архивация данных, различают следующие форматы архивных файлов: **ZIP**, **RAR**, **TER**, **ARJ**, **CAB**, **LZH**, **ACE**, **7z** и др. Чаще всего, особенно в сети Интернет, используют архивные файлы формата **ZIP**.

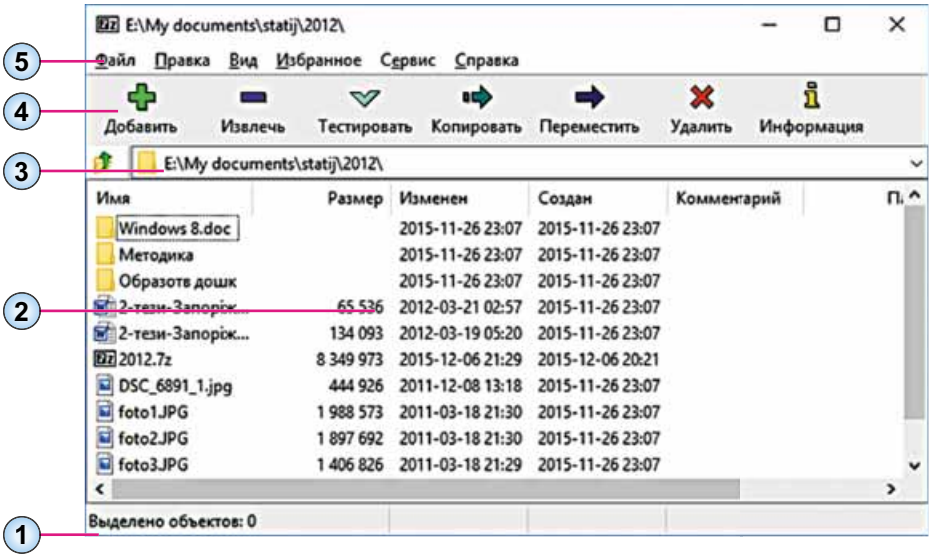
ОПЕРАЦИИ НАД АРХИВАМИ

К основным операциям над архивами относятся:

- создание архивов файлов и папок с возможным сжатием данных;
- добавление файлов и папок к уже существующим архивам;
- просмотр содержимого архивов;
- замена и обновление файлов и папок в архивах;
- извлечение из архива всех или только избранных файлов и папок;
- создание многотомных архивов (архив разбивается на несколько отдельных файлов – томов); размер томов устанавливает пользователь;
- создание обычных и многотомных архивов, которые содержат программу самостоятельного извлечения файлов и папок без участия программы-архиватора, – так называемых **SFX**-архивов (англ. ***SelF eXtracting*** – самоизвлечение);
- проверка целостности данных в архивах;
- шифрование данных и имен файлов в архивах и др.

Рассмотрим последовательность выполнения операций над архивами с использованием программы **7-zip**. Эта программа обеспечивает работу с архивными файлами типа 7z, ZIP, CAB, Z, RAR, ARJ, BZIP2, TAR, LZH и др. Особенностью является то, что программа распространяется с лицензией **freeware** и может быть свободно загружена с сайта авторов программы. Существуют версии программы как для ОС семейства **Windows**, так и для **Linux**.

Окно программы имеет вид, представленный на рисунке 2.52.



1. Строка состояния
2. Рабочее поле
3. Строка адреса
4. Панель инструментов
5. Строка меню

Рис. 2.52. Окно программы **7-zip**

Для создания архивного файла следует выполнить следующий алгоритм:

1. Запустить программу **7-zip** (***Пуск*** ⇒ ***Все приложения*** ⇒ ***7-zip***).
2. Сделать текущей папку, в которой размещены включаемые в архив файлы.
3. Выделить файлы, которые планируется включить в архив, и выбрать кнопку **Добавить**.
4. Установить в окне **Добавить к архиву** значения свойств архива (рис. 2.53):
   - *Имя* – в поле **Архив**;
   - *Место сохранения* – используя кнопку **Пролистать** ...;
   - Формат архива – *7z*, *tar*, *wim* или *zip*;
   - Уровень сжатия – *без сжатия*, *скоростной*, *быстрый*, *нормальный*, *максимальный*, *ультра*;
   - Метод сжатия;
   - *Разбить на тома размером* (*в байтах*) – при необходимости записать архивный файл на несколько носителей, имеющих сравнительно небольшой размер, например файл с видео на компакт-диск;
   - *Создать SFX-архив* – для случаев, когда необходимо будет выполнить извлечение файлов из архива без программы-архиватора;
   - *Введите пароль* и *Повторите пароль* – для защиты содержимого архива от доступа посторонних лиц

   и другие.
5. Выбрать кнопку **ОК**.

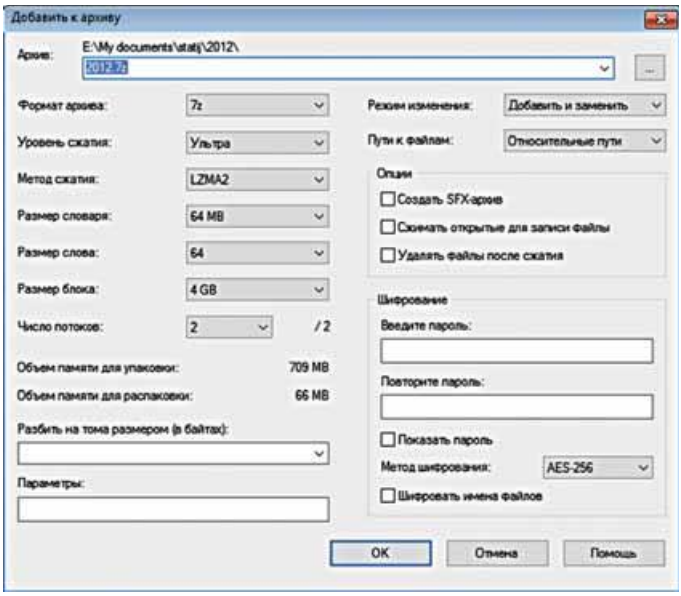


Рис. 2.53. Окно **Добавить к архиву**

Для добавления файлов в уже существующий архив следует на четвертом шаге исполнения алгоритма по созданию архива указать имя и путь к уже существующему файлу-архива.

Для просмотра содержимого архивного файла следует открыть его в окне проводника или программы-архиватора.

Для извлечения всех файлов из архива следует:

1. Запустить программу **7-zip**.
2. Сделать текущей папку, в которой размещен файл архива, и выбрать его.
3. Выбрать кнопку **Извлечь** .
4. Указать путь к папке, в которую будет выполнено извлечение, и другие значения свойств в окне **Извлечь**.
5. Выбрать кнопку **ОК**.

Для извлечения отдельных файлов и папок из архива следует:

1. Запустить программу **7-zip**.
2. Выбрать нужный архивный файл.
3. Выполнить ***Файл*** $\Rightarrow$ ***Открыть***.
4. Выделить файлы и папки, которые планируется извлекать.
5. Выбрать кнопку **Извлечь** .
6. Указать путь к папке, в которую будет выполнено извлечение, в окне **Копировать**.
7. Выбрать кнопку **ОК**.

Для удаления файлов из архива следует открыть файл архива в программе **7-zip**, выделить необходимые файлы и выбрать кнопку **Удалить** .

После правильной установки программы-архиватора в контекстное меню файлов и папок добавляются команды работы с архивами (рис. 2.54), которые можно использовать без запуска программы-архиватора.

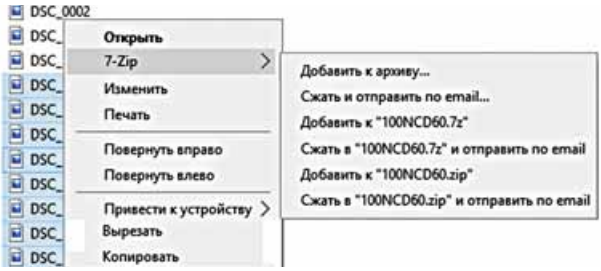


Рис. 2.54. Контекстное меню группы файлов

Для тех, кто работает с Linux

Выполнить операции с архивами в **Linux Ubuntu** можно с использованием, кроме уже описанной программы **7-zip**, программы **Менеджер архивов** . В ходе создания архивов может быть выбран один из более чем 10 форматов архивных файлов, можно установить пароль доступа к архиву и разделить архив на несколько томов.

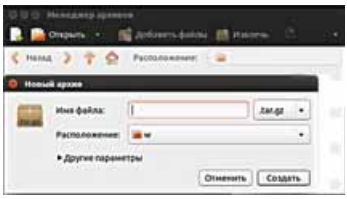


Рис. 2.55. Окно программы **Менеджер архивов**

Самое важное в этом пункте

Для уменьшения размеров данных используют **сжатие данных** по специальным **алгоритмам (методам) сжатия. Сжатие данных** – это процесс перекодировки данных с целью уменьшения размеров файлов.

Различают алгоритмы сжатия, обеспечивающие сжатие ***без потери данных***, и алгоритмы, предусматривающие ***частичную потерю данных***.

Создание копий данных с использованием специальных программ, которые могут использовать сжатие данных, называют **архивацией**. Для создания **резервных копий** и **образов системы** используются соответствующие утилиты, которые, как правило, включены в состав операционных систем.

Программы, используемые для выполнения операций над архивами, называют **архиваторами**. Над архивами выполняются следующие операции: *создание* (обычных, многотомных и SFX-архивов), *просмотр содержимого*, *добавление* новых и *замена* ранее включенных объектов, *удаление* объектов, *извлечение* всех или отдельных объектов и т. п.

Дайте ответы на вопросы

1°. Для чего используется сжатие данных?

2°. В каких случаях возможно использование сжатия с частичной потерей данных?

3°. Для чего выполняется архивация данных?

4*. Что такое архивация и что такое сжатие файлов? Какая между ними связь и какая разница?

5•. Как называют программы, выполняющие архивацию данных? Какие операции над архивами они могут выполнять?

6°. Назовите форматы архивных файлов, которые вы знаете.
7•. Какую последовательность действий необходимо выполнить при создании архива нескольких файлов?
8•. В чем отличие в действиях при извлечении всех файлов из архива и нескольких файлов из архива?
9°. Как удалить группу файлов из архива?
10•. Какую последовательность действий необходимо выполнить при создании архива нескольких файлов?
11•. Какую последовательность действий необходимо выполнить при создании архива нескольких файлов с использованием контекстного меню?

Выполните задания

1°. Запустите на выполнение программу-архиватор. Создайте архив из первых пяти файлов с расширением имени **doc**, которые зарегистрированы в папке **Архив** папки **Глава 2\пункт 2.5**. Поместите этот файл в вашу папку.
2°. Извлеките все файлы из файла **Архив1.7z** из папки **Глава 2\пункт 2.5\Архив** в папку **Мой архив**, которую нужно создать в вашей папке.
3°. Добавьте к архивному файлу **Архив2.zip**, который находится в папке **Глава 2\пункт 2.5\Архив**, два последних файла этой самой папки. Сохраните измененный архив с тем же именем в вашей папке.
4•. Запустите архиватор и:
а) создайте архив из первых пятнадцати файлов с расширением имени **doc**, которые зарегистрированы в папке **Мои документы** (**Документы** папки с именем пользователя) или из пятнадцати последних файлов с расширением имени **doc**, которые зарегистрированы в папке **Архив** папки **Глава 2\пункт 2.5**;
б) установите во время архивации следующие значения параметров: имя архива – *документы*; папка – *Рабочий стол*; формат архива – *ZIP*; метод сжатия – *без сжатия*;
в) используя контекстное меню, извлеките все файлы из созданного архива в папку **DOC**, которую создайте в папке **Мои документы** (**Документы** папки с именем пользователя).
5*. Проведите исследование по степени сжатия одних и тех же файлов при различных режимах сжатия в формате **7z**. Для исследования архивации используйте все файлы с расширением имени **bmp**, зарегистрированные в папке **Глава 2\пункт 2.5\Рисунки**. Заполните таблицу. *Примечание: коэффициент сжатия – это отношение размера файлов до сжатия к размеру файлов после сжатия.*

№ п/п	*Имя файла архива*	*Степень сжатия*	*Размер файлов*		*Коэффициент сжатия*
			до сжатия	*после сжатия*	
1	Рисунки1.7z				
2	Рисунки2.7z				

Продолжение таблицы

№ п/п	Имя файла архива	Степень сжатия	Размер файлов		Коэффициент сжатия
			до сжатия	после сжатия	
3	Рисунки3.7z				
4	Рисунки4.7z				
5	Рисунки5.7z				
6	Рисунки6.7z				

6*. Запустите архиватор и:

а) создайте архив из первых двух файлов, зарегистрированных в папке **Образцы музыки (Глава 2\пункт 2.5\Образцы музыки)**, установив при этом следующие значения параметров архивации: имя архива – *Archiv02*; формат архива – *7z*; *SFX-архив*;

б) удалите из созданного архивного файла **Archiv02.exe** последний файл;

в) добавьте к этому архиву третий файл из папки **Образцы изображений (Глава 2\пункт 2.5\Образцы изображений)**;

г) извлеките все файлы из созданного архивного файла в папку **Музыка**, которую создайте в папке **Мои документы (Документы)**.

ПРАКТИЧЕСКАЯ РАБОТА № 3

«Архивация данных и извлечение их из архивов»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Запустите на выполнение архиватор **7-zip**.
2. Создайте в папке архив из первых шестнадцати файлов с расширением имени **doc**, содержащиеся в папке **Глава 2\пункт 2.5\Архив**, установив при этом такие значения параметров архивации:
   - формат архива – *7z*;
   - степень сжатия – *быстрый*;
   - создать *SFX-архив*.
3. Выполните такие действия:
   - протестируйте архив после создания;
   - создайте в вашей папке папку **Копия архива**, скопируйте в нее архив и удалите из него последние пять файлов.
4. Определите и запишите в тетрадь:
   - общий размер файлов до архивации;
   - общий размер файлов в архиве после архивации;
   - коэффициент сжатия.
5. Используя контекстное меню архивного файла в окне **Проводника**, извлеките все файлы из созданного вами архива в папку **DOC**, которую создайте в вашей папке.
6. Используя архиватор, извлеките первые десять файлов из архива, который вы создали, в папку **1_10**, которую создайте в вашей папке.
7. Закройте все окна.

Глава 3. Обработка текстовых данных

В этой главе вы узнаете:

- о вставке в текстовый документ символов, которые отсутствуют на клавиатуре
- о создании и обработке в текстовом документе колонок, нумерованных и маркированных списков
- о вставке в текстовый документ таблиц, формул, графических изображений и их обработке
- о настройках параметров страниц текстового документа и создании колонтитулов
- о стилях и их использовании
- о создании ссылок и указателей в текстовом документе
- о просмотре структуры документа и автоматическом создании его содержания
- о шаблонах документов и их использовании
- о работе с несколькими документами одновременно

3.1. СОЗДАНИЕ И ОБРАБОТКА В ТЕКСТОВОМ ДОКУМЕНТЕ СИМВОЛОВ, КОЛОНОК, СПИСКОВ

1. Какие способы выделения текста вы знаете?
2. Какие операции редактирования символов и абзацев текста вам известны? Как они осуществляются?
3. Какие операции форматирования символов и абзацев текста вы знаете? Как они выполняются?
4. Какое назначение маркеров на горизонтальной линейке?

ВСТАВКА В ТЕКСТОВЫЙ ДОКУМЕНТ СПЕЦИАЛЬНЫХ СИМВОЛОВ

Вы уже умеете создавать и обрабатывать в текстовом процессоре **Word 2007** несложные текстовые документы: вводить текст, редактировать и форматировать символы, абзацы и страницы, печатать документ. Однако возможности текстового процессора значительно шире. Рассмотрим некоторые из них.

Так, в текстовом процессоре **Word 2007** можно вставлять в документ символы, отсутствующие на клавиатуре, – буквы других алфавитов, математические символы, условные обозначения и др. Например: © ≤ β ∑ € ∞ ≈ ╬ ☺ ♫ § Ꮷ Ϫ Ⱬ.

Для вставки в текст таких символов следует выполнить следующий алгоритм:

1. Установить курсор в нужное место текста.
2. Выбрать на вкладке **Вставка** в группе **Символы** кнопку **Символ** Ω.
3. Выбрать необходимый символ в открывшемся списке (рис. 3.1).

Если необходимый символ отсутствует в списке, то следует:

1. Выбрать команду **Другие символы**.
2. Посмотреть перечень символов в диалоговом окне **Символ** (рис. 3.2) на вкладках **Символы** и **Специальные знаки**. При необходимости можно в списке поля **Шрифт** или списке **Набор** выбрать другое значение, чтобы вывести в окне другой набор символов.
3. Выбрать необходимый символ.
4. Выбрать кнопку **Вставить**.

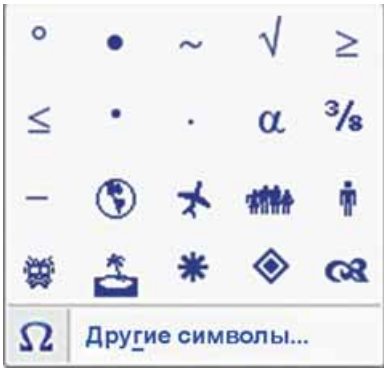


Рис. 3.1. Список символов

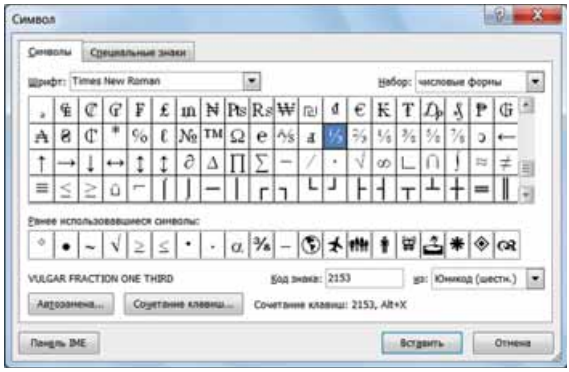


Рис. 3.2. Окно **Символ**

Символам, которые будут встречаться в тексте часто, можно в окне **Символ** назначить сочетание клавиш, при нажатии которых необходимый символ будет вставляться в текущее место текста. Для этого в окне **Символ** следует выбрать кнопку **Сочетание клавиш** и заполнить поля окна **Настройка клавиатуры**.

В текстовом процессоре **Word 2007** в документ можно вставлять и символы в виде рисунков. Например:



Для этого используют графические шрифты MS Outlook, Webdings, Wingdings и др. При нажатии на клавиши клавиатуры в текст вводятся не буквы, а разнообразные маленькие рисунки. Эти символы также можно вставить в окне **Символ**.

При редактировании текста, особенно сохраненного из Интернета, желательно включать *режим отображения скрытых символов*, выбрав на вкладке ленты **Главная** в группе **Абзац** кнопку **Отобразить все знаки** ¶.

К *скрытым* (другое название *непечатаемым*) относятся символы, которые вводятся пользователем с клавиатуры при наборе текста, но обычно не отображаются на экране и не выводятся на бумаге при печати. Но если включить режим отображения этих символов, то в соответствующих местах текста появятся специальные обозначения этих символов (табл. 3.1).

Таблица 3.1

Примеры скрытых (непечатаемых) символов

Символ	*Клавиши ввода*	*Обозначение*	*Символ*	*Клавиши ввода*	*Обозначение*
Пробел	**Пробел**	·	Неразрывный пробел	**Ctrl + Shift + пробел**	°
Конец абзаца	**Enter**	¶	Разрыв строки	**Shift + Enter**	↵
Табуляция	**Tab**	→	Разрыв страницы	**Ctrl + Enter**	·····Разрыв страницы·····

Включение режима отображения скрытых символов позволяет видеть на экране все символы текста и лучше понимать структуру документа. Так, например, если удалить символ **Конец абзаца** ¶, то этот абзац объединится со следующим. А если в любом месте абзаца установить курсор и нажать клавишу **Enter**, то абзац в этой позиции будет разделен на два и в месте разрыва появится символ **Конец абзаца** ¶. А в тех местах текста, где нажимали клавишу **Пробел**, будет отображаться скрытый символ ·, если его удалить, то слова в этом месте «слипнутся».

Редактирование текста и форматирование любых символов, вставленных в текст, осуществляются по известным вам правилам.

Для тех, кто хочет знать больше

Некоторые последовательности символов при вводе **Word 2007** автоматически заменяет на специальные символы. Примеры таких последовательностей приведены в таблице 3.2.

Таблица 3.2

Ввод специальных символов

Последовательность введенных символов	*Специальный символ*	*Последовательность введенных символов*	*Специальный символ*
слово, пробел, дефис, пробел, слово	– тире	--> минус, минус, больше	→
:)	☺	:|	😐

Режим автоматической замены некоторых последовательностей символов на специальные в **Word 2007** установлен по умолчанию. Пользователь может по собственному желанию добавить в список автозамены свои элементы, выбрав кнопку **Автозамена** в диалоговом окне **Символ**.

Для тех, кто работает с LibreOffice

Для вставки символов, которых нет на клавиатуре, в **LibreOffice Writer** следует выбрать кнопку **Специальные символы** Ω на панели инструментов **Стандартная.**

Для отображения скрытых символов следует выбрать кнопку **Непечатаемые символы** ¶ на панели инструментов **Стандартная.**

СПИСКИ И ИХ СОЗДАНИЕ В ТЕКСТОВОМ ДОКУМЕНТЕ

Особым видом форматирования абзацев текстового документа является оформление их в виде *списков*. В виде списков могут представляться перечни объектов, описания порядка действий и т. п. Например, список фамилий учеников класса, инструкция по пользованию прибором, перечень правил оформления документа, список лекарств в аптечке, последовательность действий при приготовлении какого-либо блюда.

В текстовом процессоре **Word 2007** можно создавать списки трех типов:

- ***Маркированный***, в котором каждый элемент списка в начале первой строки абзаца отмечается некоторым символом (*маркером*). Например:

Времена года: ▪ Зима ▪ Весна ▪ Лето ▪ Осень	*Времена года*: ➢ Зима ➢ Весна ➢ Лето ➢ Осень	*Времена года*: ❖ Зима ❖ Весна ❖ Лето ❖ Осень

- ***Нумерованный***, в котором в начале первой строки каждого абзаца списка указывается их номер. Порядковый номер в списке может задаваться числом, буквой или числительным и т. п. Например:

Города: 1. Запорожье 2. Киев 3. Кременчуг 4. Черновцы	*Объекты документа:* а) символ б) абзац в) страница г) документ	*Содержание учебника:* Пункт 1. Текстовый процессор Пункт 2. Табличный процессор Пункт 3. Графический редактор Пункт 4. Мультимедийные презентации

- ***Многоуровневый***, в котором в элементы списка могут быть вложены другие списки. Максимальное количество вложений элементов многоуровневого списка – 9 уровней. Например:

Времена года: 1. Зима • Декабрь • Январь • Февраль 2. Весна • Март • Апрель • Май 3. Лето 4. Осень	*Времена года*: 1. Зима а) Декабрь б) Январь в) Февраль 2. Весна а) Март б) Апрель в) Май 3. Лето 4. Осень	*Времена года*: 1. Зима 1.1. Декабрь 1.2. Январь 1.3. Февраль 2. Весна 2.1. Март 2.2. Апрель 2.3. Май 3. Лето 4. Осень

Существует несколько способов оформления фрагмента текста в виде списка.

I способ. Основной. До начала ввода списка следует разместить курсор в необходимом месте документа и выполнить ***Главная*** ⇒ ***Абзац*** ⇒ ***Маркеры*** или ***Нумерация***. В текущем месте документа появится маркер или номер того вида списка, который использовался последним. Для выбора другого вида маркера или номера следует выбрать кнопку у нужного типа списка и в открытом перечне **Библиотека маркеров** или **Библиотека нумерации** (рис. 3.3) выбрать нужный вариант оформления. После этого можно вводить первый элемент списка.

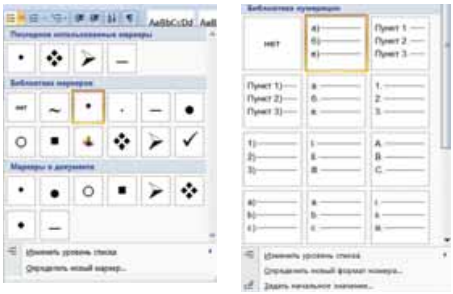


Рис. 3.3. Виды маркированных и нумерованных списков

После ввода первого элемента списка нужно нажать клавишу **Enter** – в следующей строке автоматически появятся номер или маркер. Когда последний элемент списка будет введен, нужно повторно выбрать кнопку соответствующего списка на **Ленте**, или дважды нажать **Enter**, или удалить номер или маркер клавишей **Backspace**.

II способ. Преобразование абзацев в список. Если некоторые абзацы введенного ранее текста требуется оформить как список, то следует:

1. Выделить необходимые абзацы.
2. Выбрать на **Ленте** кнопку меню соответствующего типа списка: **Маркеры** или **Нумерация**.
3. Выбрать в списке необходимый вид маркера или номера.

После этого у первой строки каждого из выделенных абзацев появится выбранный маркер или номер.

Для тех, кто хочет знать больше

III способ. Автоматическое создание списка. Разместив курсор в нужном месте документа, следует ввести некоторые символы, которые определяют желаемый вид списка (табл. 3.3), и нажать клавишу **Пробел**. Сразу же условные

символы превратятся в соответствующий маркер или номер. Далее можно начинать ввод первого элемента списка.

Таблица 3.3

Символы для создания списка

Символы	*Оформление списка*	*Символы*	*Оформление списка*
Для маркированных списков			
звездочка *	•	буква **o**	◦
минус -	–	больше **>**	➢
Для нумерованных списков			
символ 1 и точка.	1.	символ 1 и скобка	1)
символ 1 и символ больше >	1>	буква **a** и скобка	a)

Для тех, кто работает с LibreOffice

В **LibreOffice Writer** для оформления некоторого фрагмента текста в виде списка следует использовать кнопки **Маркированный список** или **Нумерация** на панели инструментов **Форматирование.** Более широкие возможности по установке свойств списков предоставляют пользователю элементы окна **Нумерация и маркировка**, которое можно открыть, выполнив ***Формат*** ⇒ ***Маркеры и нумерация*** или выбрав команду **Еще маркеры** в списке одной из указанных кнопок.

РЕДАКТИРОВАНИЕ СПИСКОВ

В текстовом процессоре **Word 2007** созданные списки удобно редактировать.

Если в любом месте списка нужно прибавить еще одну строку, то после предыдущего элемента списка следует нажать клавишу **Enter** – будет вставлен дополнительный абзац с номером или маркером, а нумерация последующих элементов списка автоматически изменится.

Для того чтобы удалить элемент списка, нужно выделить требуемый абзац и нажать клавишу **Delete** или **Backspace** – элемент списка будет удален, нумерация автоматически изменится.

Иногда необходимо в некотором месте текста отменить нумерацию, например, для ввода текста, который не является элементом списка. В таком случае нумерацию следует удалить повторным выбором кнопки списка на **Ленте** или нажатием **Backspace.**

При копировании и перемещении фрагментов нумерованных списков автоматическая нумерация может не соответствовать потребности пользователя. Для изменения нумерации элементов списка следует:

1. Выбрать указателем мыши место в начале строки перед номером, который нужно изменить. При этом нумерация элементов списка выделится.
2. Открыть контекстное меню выделенного фрагмента (рис. 3.4).

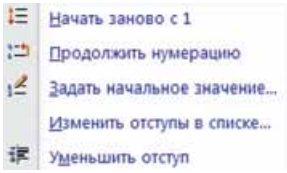


Рис. 3.4. Контекстное меню номера списка

3. Выбрать нужный вариант изменения номера:
   - **Начать заново с 1** – нумерация элементов списка начнется с первого номера.
   - **Продолжить нумерацию** – нумерация последующих элементов списка будет продолжена дальше от предыдущего номера.
   - **Задать начальное значение** – нумерация элементов списка начнется с номера, который укажет пользователь.

Элементы списка можно отсортировать в алфавитном, числовом или хронологическом порядке (по убыванию или по возрастанию). При этом нумерация элементов останется последовательной, а абзацы текста будут переставлены в соответствии с выбранным порядком.

Для этого нужно выполнить следующий алгоритм:

1. Выделить абзацы, которые упорядочиваются.
2. Выполнить ***Главная*** ⇒ ***Абзац*** ⇒ ***Сортировка*** .
3. Установить в диалоговом окне **Сортировка текста** (рис. 3.5) следующие значения:
   - **Сортировать по** – *абзацам*.
   - **Тип** данных – *текст*, *число* или *дата*.
   - **Порядок сортировки** – *по возрастанию* или *по убыванию*.
4. Выбрать кнопку **ОК**.

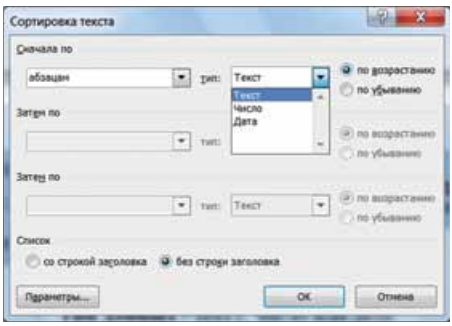


Рис. 3.5. Окно **Сортировка текста**

ФОРМАТИРОВАНИЕ СПИСКОВ

При необходимости пользователь может отформатировать созданный список: изменить вид маркера, его формат, способ нумерации, способ выравнивания списка, отступы текста от маркера и номера и др.

Для этого следует выделить нужные элементы списка и выполнить форматирование известными средствами – использовать элементы управления мини-панели вкладки **Главная**, диалоговых окон и др. Также можно использовать маркеры на горизонтальной линейке для изменения отступов текста списка от полей (рис. 3.6).

Для установки отступа текста списка от номера или маркера используется **табуляция** – средство, которое дает возможность размещать объекты в строке в строго определенных местах **(позициях табуляции)**. На линейке позиция табуляции, по умолчанию, отмечается меткой . Как видно на рисунке 3.6 слева, метка табуляции стоит на отметке 1 см, а справа –

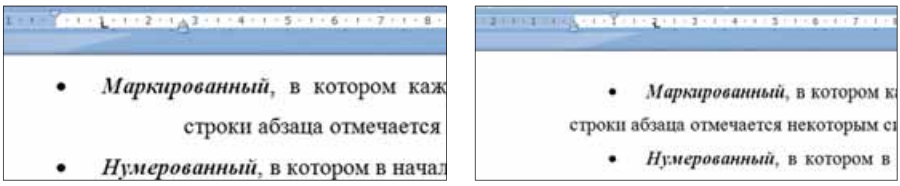


Рис. 3.6. Маркеры и метка табуляции на линейке

на отметке 2 см. Поэтому текст элементов списка начинается именно с таким отступом.

Для установки метки табуляции следует на линейке выбрать нужное место. Чтобы изменить позицию табуляции, нужно перетащить соответствующую метку вдоль линейки в новое место. Для удаления позиции табуляции достаточно переместить метку табуляции за пределы линейки.

Также установить и изменить позицию табуляции можно в окне **Табуляция**, для чего следует выполнить ***Главная*** ⇒ ***Абзац*** ⇒ ***Табуляция***.

Одной из особенностей форматирования списков является то, что формат маркеров (номеров) и формат символов текста списка может быть разным. Если выбрать один из маркеров или номеров списка, то будут выделены все аналогичные объекты во всем списке. Далее можно изменять их формат независимо от формата символов текста списка (рис. 3.7).


1. Понедельник
2. Вторник
3. Среда
4. Четверг
5. Пятница
6. Суббота
7. Воскресенье

1. Понедельник
2. Вторник
3. Среда
4. Четверг
5. Пятница
6. Суббота
7. Воскресенье

Рис. 3.7. Пример форматирования списка

При необходимости пользователь может создать собственный формат оформления списка, выбрав команду **Определить новый маркер** или **Определить новый формат номера** внизу диалоговых окон соответствующих списков (см. рис. 3.3). В открывшемся окне необходимо установить значения свойств нового вида списка.

Для тех, кто хочет знать больше

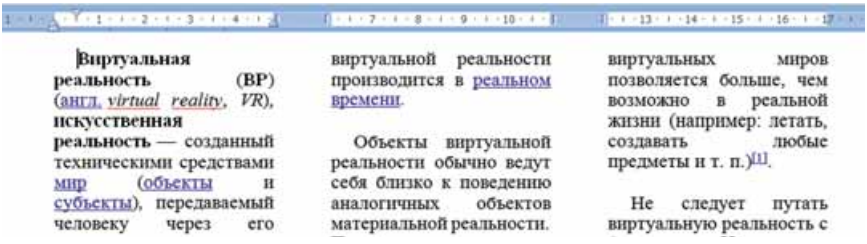
Многоуровневые списки

Для создания многоуровневых списков следует выбрать на **Ленте** кнопку **Многоуровневый список** . Чтобы выбрать вид многоуровневого списка, нужно открыть список команд этой кнопки и выбрать нужный вид списка в разделе **Библиотека списков**.

Создание многоуровневого списка может начинаться с оформления одноуровневого списка (маркированного или нумерованного). Превращение его в многоуровневый осуществляется изменением уровня вложенности некоторых элементов списка с помощью кнопок **Уменьшить отступ** (переход на уровень выше) или **Увеличить отступ** (переход на уровень ниже), которые расположены на **Ленте** в группе **Абзац**.

СОЗДАНИЕ КОЛОНОК В ТЕКСТОВОМ ДОКУМЕНТЕ

Текстовый процессор **Word** позволяет размещать текст документа в несколько колонок (рис. 3.8). Как это делают, например, в газетах и журналах.


Виртуальная реальность (ВР) (англ. *virtual reality*, *VR*), **искусственная реальность** — созданный техническими средствами мир (объекты и субъекты), передаваемый человеку через его

виртуальной реальности производится в реальном времени.

Объекты виртуальной реальности обычно ведут себя близко к поведению аналогичных объектов материальной реальности.

виртуальных миров позволяется больше, чем возможно в реальной жизни (например: летать, создавать любые предметы и т. п.)[1].

Не следует путать виртуальную реальность с

Рис. 3.8. Пример текста, размещенного в три колонки

Чтобы некоторый фрагмент текста разместить в колонках, его нужно выделить и выполнить последовательность действий ***Разметка страницы*** ⇒ ***Параметры страницы*** ⇒ ***Колонки***. Затем в списке следует выбрать нужное количество колонок (рис. 3.9). Текст автоматически будет размещен выбранным способом.

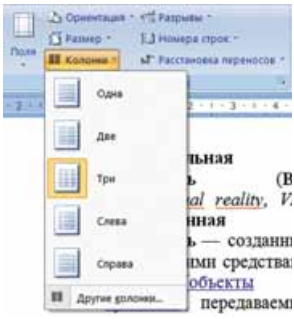


Рис. 3.9. Открытый список кнопки **Колонки**

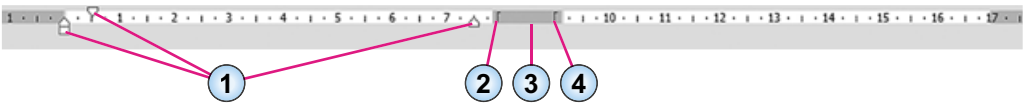
Чтобы отменить расположение фрагмента текста в колонках, этот фрагмент нужно выделить и установить для него режим отображения в одну колонку. При этом перед первой строкой и после последней строки фрагмента текста останутся скрытые символы **Разрыв раздела**, которые желательно удалить.

Для тех, кто работает с LibreOffice

В **LibreOffice Writer** для оформления некоторого фрагмента текста в несколько колонок нужно выделить этот фрагмент, выполнить ***Формат*** ⇒ ***Колонки*** и в окне **Колонки** установить значения свойств колонок с текстом.

РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ КОЛОНОК

Когда курсор находится в тексте, размещенном в колонках, на горизонтальной линейке, кроме маркеров абзаца, появляются *маркеры границ* колонок (рис. 3.10), перетаскивая которые можно менять ширину колонок и расстояние между ними.



1. Маркеры отступов абзацев
2. Маркер правой границы первой колонки
3. Промежуток между колонками
4. Маркер левой границы второй колонки

Рис. 3.10. Маркеры границ колонок на горизонтальной линейке

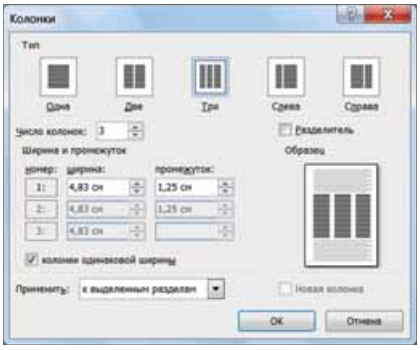


Рис. 3.11. Окно **Колонки**

Форматирование колонок можно выполнить и в диалоговом окне **Колонки** (рис. 3.11), которое открывается ***Разметка страницы*** ⇒ ***Параметры страницы*** ⇒ ***Колонки*** ⇒ ***Другие колонки***. Здесь можно установить количество колонок, ширину каждой из них, расстояние между отдельными колонками, наличие разделителя (вертикальная линия между колонками). После установки значений свойств колонок нужно выбрать кнопку **ОК**.

Самое важное в этом пункте

Текстовый процессор **Word 2007** позволяет вставлять в текст символы, отсутствующие на клавиатуре. Для этого используют список символов, который открывается выбором одноименной кнопки на вкладке **Вставка** в группе **Символы**. Также в текст можно вставить различные маленькие рисунки, выбрав для этого специальные шрифты MS Outlook, Webdings, Wingdings и др.

К *скрытым* (*непечатаемым*) символам относятся символы, которые вводятся пользователем с клавиатуры, не выводятся на бумагу при печати и, как правило, не отражаются на экране, – пробел, конец абзаца, табуляция и др. Для отображения этих символов в тексте следует выполнить ***Главная*** ⇒ ***Абзац*** ⇒ ***Показать все знаки*** ¶.

Абзацы текста можно оформить в виде маркированных, нумерованных или *многоуровневых списков*, которые используются для наглядного представления перечня объектов. Для этого следует выделить фрагмент текста с нужными абзацами и в меню кнопок **Маркеры** или **Нумерация** выбрать нужный вид списка.

При редактировании списков можно добавлять новые элементы списка, удалять лишние, сортировать по возрастанию или убыванию. Под форматированием списка понимают установку отступов маркеров от поля, отступов текста от маркеров, изменение формата номеров и видов маркеров, начального номера списка и т. п.

Также фрагмент текста можно разместить в несколько колонок. Для этого следует выделить нужный текст, выполнить ***Разметка страницы*** ⇒ ***Параметры страницы*** ⇒ ***Колонки*** ⇒ ***Другие колонки***, установить в окне **Колонки** нужное количество колонок, ширину каждой из них, расстояние между отдельными колонками, наличие разделителя.

Дайте ответы на вопросы

1•. Как вставить в текст символы, отсутствующие на клавиатуре (математические, латинские, графические)? Как осуществляется их редактирование и форматирование?

2•. Какие символы называют скрытыми (непечатаемыми)? Приведите примеры таких символов. Для чего их используют?

3°. Списки каких типов можно создать в текстовом документе **Word 2007**?

4°. Какое название и назначение кнопок , , группы **Абзац**?

5•. В чем заключается автоматическая нумерация элементов списка? Как изменить номер элемента списка?

6°. Как отменить нумерацию (маркировку) после окончания ввода элементов списка?

7•. В списке учеников вашего класса было пропущено несколько фамилий. Как вставить их в список?

8•. Список учеников вашего класса был введен в произвольном порядке. Как разместить фамилии в списке в алфавитном порядке?

9•. Как разместить текст в несколько колонок? Какие свойства колонок можно изменять?

10•. Как можно установить ширину колонок, отступы между колонками, количество колонок?

Выполните задания

1°. Откройте программу **Word 2007** и введите по 5 разных графических символов, используя графические шрифты MS Outlook, Webdings, Wingdings. Сохраните документ в файле с именем **задание 3.1.1.docx** в вашей папке.

2°. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.1\образец 3.1.2.docx**). Первый – третий абзацы документа оформите как маркированный список (маркер выберите самостоятельно), абзацы четвертый – шестой – как нумерованный список (вид нумерации выберите самостоятельно). Отмените нумерацию для заголовка текста. Сохраните документ в файле с именем **задание 3.1.2.docx** в вашей папке.

3•. Создайте новый документ и введите список фамилий ваших одноклассников в виде нумерованного списка. Упорядочьте их по алфавиту. Сохраните документ в вашей папке в файле с именем **задание 3.1.3-1.docx**. Преобразуйте список в маркированный. Сохраните документ в вашей папке с именем **задание 3.1.3-2.docx**.

4•. Создайте новый документ и введите перечень семи чудес Украины (сведения найдите в Интернете) в виде маркированного списка. Установите следующие значения свойств: отступ маркера от левого поля – *2 см*, отступ текста элементов списка от маркера – *1 см*. Сохраните документ в вашей папке в файле с именем **задание 3.1.4.docx**.

5•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.1\образец 3.1.5. docx**). Оформите документ по образцу, представленному в файле. Сохраните документ в вашей папке в файле с именем **задание 3.1.5.docx**.

6*. Выясните, используя **Справку**, в каком случае маркеры и номера могут не появляться при создании списка. Как это исправить?

7*. Найдите на сайте **Microsoft Office Online** графические маркеры, с помощью **Справки** выясните, как их установить на своем компьютере, и выполните это. Оформите с их использованием список областей Украины. Сохраните документ в вашей папке в файле с именем **задание 3.1.7.docx**.

8•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.1\образец 3.1.8.docx**). В открытом документе разместите текст второго абзаца в две колонки одинаковой ширины без разделителя. Текст четвертого абзаца – в три колонки разной ширины с разделителем. Сохраните документ в вашей папке в файле с именем **задание 3.1.8.docx**.

3.2. ТАБЛИЦЫ В ТЕКСТОВЫХ ДОКУМЕНТАХ

1. Приведите примеры использования таблиц в изучении различных школьных предметов и в повседневной жизни.
2. Как переместить курсор в табличном процессоре?
3. Какие свойства имеет графическое изображение в текстовом документе? Какие значения они могут принимать?

ТАБЛИЦЫ И ИХ СВОЙСТВА

Для упорядоченного и наглядного представления в документах различных данных используют таблицы. Данные, представленные в таблице, выглядят более компактнее и удобны для восприятия (например, табл. 3.4).

Таблица 3.4

Расписание уроков в 8-А классе

№	Понедельник	Вторник	Среда	Четверг	Пятница
1	Информатика I гр. Информатика II гр.	Физическая культура		Геометрия	Иностранный язык (англ.)
2	Физика	Украинский язык	Русский язык		История Украины
3	Физическая культура	Украинская литература	Алгебра	Всемирная история	Биология
10:45–11:15	 **Динамический перерыв** 				
4	Химия	Биология	Химия	География Основы здоровья	Украинская литература
5	Алгебра	Иностранный язык (англ.)	Физика	Зарубежная литература	Украинский язык
6	История Украины	География	Информатика I гр. Информатика II гр.	Трудовое обучение 2 урока	Физическая культура
14:00 кружки					

Таблица состоит из *столбцов* и *строк*, на пересечении которых находятся *ячейки*. Столбцы, строки, ячейки являются объектами таблицы. Таблица в **Word 2007** может содержать до 63 столбцов и произвольное количество строк.

Как видно из приведенного примера (табл. 3.4), высота строк и ширина столбцов таблицы могут быть разными. Несколько ячеек могут быть объединены в одну, а некоторые могут быть разделены на несколько. Направление текста в ячейке может быть горизонтальное или вертикальное. Для разных объектов таблицы можно установить границы и заливку различного типа.

Таблица, как объект текстового документа, имеет следующие свойства:

- **размер таблицы** – определяет ширину таблицы в *сантиметрах* или в *процентах* от ширины страницы;
- **ширина столбцов, высота строк, размеры ячейки** – определяется в *сантиметрах* или *процентах* от ширины страницы;
- **способ обтекания таблицы текстом** – может принимать следующие значения: *без обтекания*, *с обтеканием вокруг таблицы*;
- **границы таблицы** – определяются *цветом*, *типом* и *шириной* границ всей таблицы или отдельных ее объектов;
- **заливка объектов таблицы** – определяется *цветом*, *узором* и т. п.

СОЗДАНИЕ ТАБЛИЦЫ

В **Word 2007** существует несколько способов создания таблицы в текстовом документе:

1. Вставить таблицу простой структуры.
2. Разработать таблицу произвольной структуры.
3. Превратить фрагмент текста в таблицу.
4. Вставить таблицу из коллекции шаблонов.

Все команды создания таблиц находятся в списке кнопки **Таблица** группы **Таблицы** вкладки **Вставка**. Рассмотрим два способа, которые используют чаще всего.

I с п о с о б. Вставить в документ таблицу простой структуры можно следующим образом:

1. Выбрать в документе место, где нужно вставить таблицу.
2. Выбрать на вкладке **Вставка** в группе **Таблицы** кнопку **Таблица**  .
3. Выделить на схеме нужное количество строк и столбцов и щелкнуть левую кнопку мыши. Например, слева на рисунке 3.12 выделена таблица, состоящая из 3 столбцов и 4 строк.

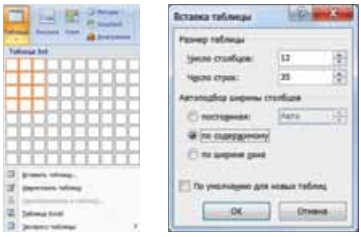


Рис. 3.12. Способы вставки таблицы

Таким способом можно вставить в документ таблицу, в которой не более 10 столбцов и 8 строк. Если же нужно создать большую таблицу, то следует выполнить ***Вставка*** ⇒ ***Таблицы*** ⇒ ***Таблица*** ⇒ ***Вставить таблицу.*** Затем в соответствующих полях диалогового окна **Вставка таблицы** (рис. 3.12, справа) нужно указать количество столбцов и строк, способ определения ширины столбцов новой таблицы и выбрать кнопку **ОК**.

II способ. Если таблица имеет сложную структуру, например как на рисунке 3.13, то ее можно *начертить* в документе. Для этого следует:

1. Выполнить ***Вставка*** ⇒ ***Таблицы*** ⇒ ***Таблица*** ⇒ ***Нарисовать таблицу***.
2. Нарисовать указателем, который примет вид карандаша, контур всей таблицы.
3. Нарисовать линии, разделяющие строки и столбцы.

Если в процессе чертежа таблицы были созданы лишние или ошибочные линии, их можно «стереть», использовав инструмент **Ластик**. Для этого следует выбрать на вкладке **Конструктор** в группе **Нарисовать границы** кнопку **Ластик**, навести указатель (его вид при этом изменится) на лишнюю линию и выбрать ее. После удаления лишних линий кнопку **Ластик** следует отключить, выбрав ее повторно.

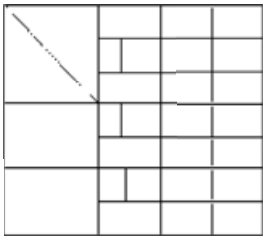


Рис. 3.13. Таблица сложной структуры

Для тех, кто хочет знать больше

III способ. Выделенный фрагмент текста можно превратить в таблицу, выполнив последовательность действий ***Вставка*** ⇒ ***Таблицы*** ⇒ ***Таблица*** ⇒ ***Преобразовать в таблицу***. В диалоговом окне указать нужное количество столбцов и символы, также нужно указать символ, который в выделенном фрагменте следует считать разделителями ячеек (пробел, табуляция, разрыв абзаца, точка с запятой и др.).

Для тех, кто работает с LibreOffice

Для вставки простой таблицы в **LibreOffice Writer** нужно выбрать кнопку **Вставить таблицу** на панели инструментов **Стандартная** и выделить на схеме нужное количество строк и столбцов, затем щелкнуть левую кнопку мыши. Таким способом можно вставить в документ таблицу, в которой не более 10 столбцов и 15 строк.

Для вставки в документ таблицы большего размера следует выполнить ***Таблица*** ⇒ ***Вставить таблицу***. Затем в соответствующих полях диалогового окна **Вставить таблицу** (рис. 3.14) нужно указать количество столбцов и строк, а также другие значения свойств и выбрать кнопку **Вставить**.

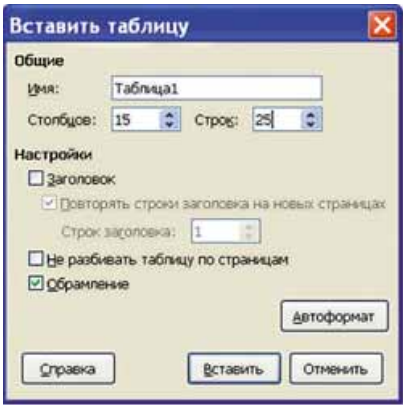


Рис. 3.14. Окно **Вставить таблицу в LibreOffice Writer**

ВВОД ДАННЫХ В ТАБЛИЦУ И ПЕРЕМЕЩЕНИЕ ПО ТАБЛИЦЕ

После того как таблица создана, ее нужно заполнить данными. Текст вводится в текущую ячейку таблицы по известным вам правилам ввода текста.

При вводе данных в ячейки ширина столбца и высота строки автоматически изменяются, потому что такой режим установлен в программе по умолчанию.

Чтобы переместить курсор в определенную ячейку, ее нужно выбрать указателем или воспользоваться клавишами управления курсором (табл. 3.5).

Таблица 3.5

Перемещение курсора в таблице

Клавиша	*Назначение*	*Клавиша*	*Назначение*
Tab	Следующая ячейка таблицы	**Shift + Tab**	Предыдущая ячейка таблицы
→	Следующий символ текста в ячейке или ячейка	←	Предыдущий символ текста в ячейке или ячейка
↓	Следующая строка текста в ячейке или таблице	↑	Предыдущая строка текста в ячейке или таблице

ВЫДЕЛЕНИЕ ОБЪЕКТОВ ТАБЛИЦЫ

Текст в ячейках таблицы, саму таблицу и ее объекты можно редактировать и форматировать. Для выполнения этих операций объекты таблицы, над которыми выполняются действия, нужно сделать *текущими* или *выделить*. Выделение объектов можно выполнить одним из двух способов:

1. Используя манипулятор мышь:
   - *Для выделения одной ячейки таблицы* – выбрать внутреннюю область ячейки у ее левой границы, когда указатель приобретает вид .
   - *Для выделения одной строки* – выбрать внешнюю область строки таблицы у ее левой границы, когда указатель приобретает вид .
   - *Для выделения одного столбца* – выбрать внешнюю область столбца таблицы у ее верхней границы, когда указатель приобретает вид .
   - *Для выделения всей таблицы* – выбрать маркер над левым верхним углом таблицы, когда указатель приобретает вид .
   - *Для выделения нескольких смежных объектов таблицы* – выделить область, в которую попадают нужные объекты таблицы.
   - *Для выделения нескольких несмежных объектов таблицы* – выделить один объект, затем, удерживая нажатой клавишу **Ctrl**, выделить остальные нужные объекты.
2. Используя элементы управления **Ленты**: сделать нужный объект таблицы текущим, а затем выполнить ***Работа с таблицами*** ⇒ ***Макет*** ⇒ ***Таблица*** ⇒ ***Выделить*** и выбрать в списке нужную команду: **Выделить ячейку, Выделить столбец, Выделить строку** или **Выделить таблицу.**

Для снятия выделения объекта нужно выбрать место за пределами выделения.

Обратите внимание, при выборе любого объекта таблицы на **Ленте** появляется две временные вкладки **Конструктор** и **Макет** во временном разделе **Работа с таблицами.**

РЕДАКТИРОВАНИЕ ТАБЛИЦЫ

Редактирование текста в таблице осуществляется обычными для **Word** способами. Кроме того, для быстрой очистки содержимого всей таблицы или ее объектов достаточно их выделить и нажать клавишу **Delete**: данные будут удалены, а сама таблица останется.

Редактирование самой таблицы предусматривает добавление или удаление отдельных ее объектов, объединение или разделение ячеек таблицы и т. п.

Добавлять пустые строки в конце таблицы можно с помощью клавиши **Tab**. Для этого нужно поставить курсор в последнюю ячейку таблицы и нажать эту клавишу. Если дополнительная строка нужна внутри таблицы, то курсор следует поставить в конце строки, за пределами таблицы и нажать клавишу **Enter**.

Удалить любую выделенную строку или столбец таблицы можно, нажав клавишу **Backspace**.

Все элементы управления для выполнения операций редактирования таблицы находятся на вкладке **Макет** (рис. 3.15). В группе **Строки и столбцы** размещены кнопки для удаления и вставки соответствующих объектов. Эти действия применяются к выделенным или текущим объектам таблицы.



Рис. 3.15. Вкладка **Макет**

Также для выполнения операций редактирования можно воспользоваться командами контекстного меню выделенных объектов таблицы.

Всю таблицу, как единый объект текстового документа, можно перемещать и копировать. Для этого таблицу нужно выделить и применить известные вам способы: перетаскивание, сочетание «горячих» клавиш или элементы управления на **Ленте**.

Для тех, кто хочет знать больше

Если несколько ячеек таблицы, которые расположены рядом, нужно *объединить* в одну, то их следует выделить и выбрать на вкладке **Макет** в группе **Объединить** кнопку **Объединить ячейки** . При этом текст ячеек «склеивается» и размещается в объединенной ячейке.

Для *разделения* одной или нескольких смежных ячеек на несколько следует выполнить:

1. Выделить нужные ячейки таблицы.
2. Выбрать кнопку **Разбить ячейки** на вкладке **Макет** в группе **Объединить**.
3. Указать в диалоговом окне, на сколько строк и столбцов следует разделить ячейки.
4. Указать способ размещения текста после разделения, используя соответствующий флажок.

Разделить ячейки или объединить ячейки можно также, использовав инструменты **Нарисовать таблицу** и **Ластик** группы **Нарисовать границы** вкладки **Конструктор**.

Когда таблицу в документе нужно разделить на две для ввода между ними текста или переноса части таблицы на другую страницу, используют кнопку

Разбить таблицу группы **Объединить** вкладки **Макет**. Для этого курсор должен находиться в той строке таблицы, где планируется разрыв.

Если таблица большая и занимает больше одной страницы текстового документа, то заголовки столбцов на других страницах можно вставить автоматически. Для этого следует выделить строки заголовка в начале таблицы и выполнить ***Макет*** ⇒ ***Данные*** ⇒ ***Повторить строки заголовков***.

Для удаления таблицы из документа и представления ее содержимого в виде обычного текста можно выполнить автоматическое преобразование таблицы в текст. Для этого нужно выделить таблицу, выполнить ***Макет*** ⇒ ***Данные*** ⇒ ***Преобразовать в текст*** и в диалоговом окне **Преобразование в текст** указать символ, который следует вставлять в местах бывшего разделения текста на отдельные ячейки.

ФОРМАТИРОВАНИЕ ТАБЛИЦЫ

Под форматированием таблицы понимают следующие операции: установка ширины столбцов и высоты строк, способа выравнивания таблицы на странице документа и текста в ячейках, направления текста в ячейке, цвета и толщины границ ячеек, заливки в ячейках и т. п.

Для форматирования таблицы предназначены элементы управления временных вкладок **Макет** (табл. 3.6) и **Конструктор** раздела **Работа с таблицами** на **Ленте**. Для форматирования текста в ячейках можно применять стандартные средства **Word**.

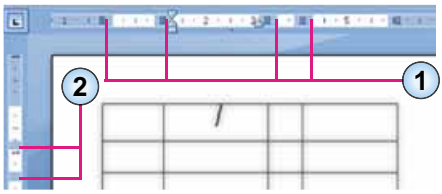
Таблица 3.6

Назначение некоторых элементов управления вкладки Макет

Элемент управления	*Назначение*
Список **Выравнивание**	Для установления способа выравнивания текста в ячейке – *снизу по левому краю*, *по центру*, *по центру по правому краю* и т. п.
Кнопка **Направление текста**	Для установления способа расположения текста в ячейке – *горизонтальное* или *вертикальное*
Кнопка **Поля ячейки**	Для открытия диалогового окна, в котором можно установить *размер полей* в ячейках и *интервал между ячейками* в таблице
Счетчики **Ширина столбца** **Высота строк**	Для установления точных размеров отдельных объектов таблицы

Размеры отдельных объектов таблицы можно изменить и другими способами:

- ***Перетаскивание границ объекта.*** Наведение указателя на границу строки или столбца в пределах таблицы меняет его вид: на границе строки указатель будет иметь вид ↕, на границе столбца – ↔. При перетаскивании штриховая линия будет демонстрировать новое положение границы объекта.



1. Маркеры границ столбцов
2. Маркеры границ строк

Рис. 3.16. Границы строк и столбцов таблицы на линейках

- ***Перетаскивание маркера границы.*** Когда курсор находится в области таблицы, на горизонтальной линейке появляются *маркеры границ столбцов*, а на вертикальной – *маркеры границ строк* (рис. 3.16). При их перетаскивании по линейке изменяются размеры соответствующих столбцов и строк.

Для границ таблицы или ее отдельных объектов можно установить цвет, толщину, стиль линий. Внутренние части объектов таблицы можно залить разными цветами, используя различные способы заливки. По умолчанию в **Word 2007** таблице назначается обрамление типа «сетка» – все ячейки имеют границы в виде тонких сплошных линий шириной 0,5 пт без заливки. Изменить значение этих свойств можно, используя элементы управления временной вкладки **Конструктор** (рис. 3.17).



Рис. 3.17. Вкладка **Конструктор**

Для выбора ширины, типа, цвета линий границ предназначены соответствующие элементы управления группы **Нарисовать границы.** В группе **Стили таблиц** можно установить цвет заливки объектов таблицы (кнопка **Заливка**), отобразить или снять отображение тех или иных границ (кнопка **Границы**). Установку значений этих свойств можно выполнить и в диалоговом окне группы **Нарисовать границы.**



Рис. 3.18. Диалоговое окно **Свойства таблицы**

Установку значений, описанных выше, и других свойств таблиц, таких как способ расположения таблицы на странице, способ обтекания таблицы текстом и т. п., можно выполнить на вкладках диалогового окна **Свойства таблицы** (рис. 3.18). Открыть это окно можно соответствующей командой контекстного меню таблицы или кнопкой **Свойства** группы **Таблица** вкладки **Макет.**

Для тех, кто хочет знать больше

Также для форматирования таблицы можно применить один из встроенных стилей, которые устанавливают определенные значения параметров форматирования объектов таблицы. Стиль можно выбрать в списке группы **Стили таблиц** на вкладке **Конструктор**. Выбор стиля сопровождается динамичным предварительным просмотром будущего результата форматирования.

Используя флажки группы **Параметры стилей таблиц** вкладки **Конструктор**, в примененном стиле можно изменить форматы отдельных частей таблицы (строки заголовка и итогов, первого и последнего столбцов, линий строк и столбцов).

Для тех, кто работает с LibreOffice

Редактирование и форматирование объектов таблицы осуществляется с использованием меню **Таблица** или элементов управления панели инструментов **Таблица** (рис. 3.19), которая появляется на экране при установлении курсора в пределах таблицы.



Рис. 3.19. Панель инструментов **Таблица**

Самое важное в этом пункте

Для упорядочения и наглядного представления в документах различных данных используют таблицы. Таблица состоит из столбцов и строк, на пересечении которых находятся ячейки. Столбцы, строки, ячейки являются объектами таблицы. В ячейках таблицы могут размещаться текст, числа, рисунки, формулы и даже другие таблицы.

Таблица, как объект текстового документа, имеет следующие свойства: размер таблицы, ширина столбцов, высота строк, способ выравнивания таблицы, способ обтекания таблицы текстом, границы таблицы, заливка объектов таблицы и т. п.

Есть несколько способов вставить в текстовый документ таблицу. Наиболее общий из них такой: выполнить ***Вставка*** ⇒ ***Таблицы*** ⇒ ***Таблица*** ⇒ ***Вставить таблицу***, затем в соответствующих полях диалогового окна **Вставка таблицы** указать количество столбцов и строк, способ определения ширины столбцов новой таблицы и выбрать кнопку **ОК**.

Редактирование самой таблицы предусматривает добавление или удаление отдельных ее объектов, объединение или разбитие ячеек таблицы и т. п. Под форматированием таблицы понимают следующие операции: установка ширины столбцов и высоты строк, способа выравнивания таблицы на странице и текста в ячейках, направления текста в ячейке, цвета и толщины границ ячеек, заливки в ячейках и т. п.

Все элементы управления для выполнения операций редактирования и форматирования таблицы находятся на вкладке **Макет** и **Конструктор** временного раздела **Работа с таблицами** на **Ленте**.

Дайте ответы на вопросы

1°. Для чего в документе используют таблицы? Из чего они состоят?
2•. Какие свойства таблицы и какие их возможные значения?
3°. Какими способами можно вставить таблицу в текстовый документ?
4•. Что понимают под термином «таблица сложной структуры»? Каким способом ее удобнее создать?
5•. Как выделить различные объекты таблицы?
6•. Как можно удалить или вставить строки или столбцы в таблицу?
7*. В чем будет заключаться отличие нажатия клавиш **Delete** и **Backspace** при выделенном фрагменте таблицы?
8•. В конце таблицы нужно добавить еще одну строку. Как это сделать быстро? Как поступить, чтобы вставить строку внутрь таблицы?

9°. Как изменить размеры столбцов или строк?

10*. Какие направления текста в ячейке можно установить? Как это сделать?

11•. Как будет размещен текст в ячейке, если выбрать кнопку  ?

12•. В каких случаях нужно объединять ячейки? Когда пользователю нужно разбить ячейки, как это можно сделать?

13°. Где находятся элементы управления для форматирования таблиц?

Выполните задания

1•. Создайте таблицу по приведенному образцу. Сохраните документ в файле с именем **задание 3.2.1.docx** в вашей папке.

НАЧИСЛЕНИЕ ЗАРАБОТНОЙ ПЛАТЫ

Фамилия	1 квартал	2 квартал	3 квартал	4 квартал
Иванов	2200.35	2200.35	2200.35	2200.35
Петров	3378.89	3378.89	3378.89	3378.89
Сидоров	5456.90	5456.90	5456.90	5456.90
Козлов	1456.67	1456.67	1456.67	1456.67
Андреев	4589.00	4589.00	4589.00	4589.00

2•. Создайте в текстовом документе расписание уроков в вашем классе по образцу таблицы 3.4. Сохраните документ в вашей папке в файле с именем **задание 3.2.2.docx** и отправьте файл по электронной почте учителю.

3•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.2\образец 3.2.3.docx**). Удалите последний столбец и последнюю строку таблицы. Вставьте перед первым столбцом дополнительный столбец и введите в его ячейки номера строк. Вставьте перед первой строкой еще одну строку. Объедините все ячейки строки и введите в ячейку название таблицы. Сохраните документ в вашей папке в файле с именем **задание 3.2.3.docx**.

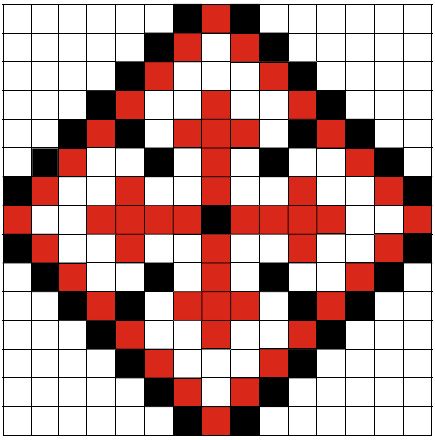
4•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.2\образец 3.2.4.docx**). Отформатируйте таблицу по указанным значениям свойств. Сохраните документ в вашей папке в файле с именем **задание 3.2.4.docx**.

Объект таблицы	Шрифт	Размер шрифта	Цвет символов	Начертание	Выравнивание
Первая строка	Comic Sans MS	14	Темно-зеленый	Полужирный	По центру
Четвертая строка	Arial	12	Темно-синий	Курсив	По левому краю
Первый столбец	Times New Roman	14	Черный	Обычный	По правому краю

5•. Создайте в текстовом документе, используя таблицу, схему классификации треугольников по равенству сторон. Сохраните документ в вашей папке в файле с именем **задание 3.2.5.docx.**



6•. Создайте в текстовом документе таблицу, закрасив ее ячейки по приведенному образцу. Сохраните документ в вашей папке в файле с именем **задание 3.2.6.docx.**



7*. Создайте таблицу-календарь на текущий месяц. Сохраните документ в вашей папке в файле с именем **задание 3.2.7.docx.**

8*. Вы создали новый документ, в котором с самого начала страницы вставили таблицу и забыли написать название таблицы. Выясните с помощью **Справки**, как вставить текст перед таблицей. Запишите алгоритм в тетрадь.

3.3. СОЗДАНИЕ, РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ ГРАФИЧЕСКИХ ОБЪЕКТОВ В ТЕКСТОВОМ ДОКУМЕНТЕ

1. Назовите свойства графического изображения в текстовом документе **Word 2007**. Какие значения они могут принимать?
2. Назовите известные вам операции редактирования графических изображений в **Word 2007**. Как они выполняются?
3. Назовите известные вам операции форматирования графических изображений в **Word 2007**. Как они выполняются?

ГРАФИЧЕСКИЕ ОБЪЕКТЫ В ТЕКСТОВОМ ДОКУМЕНТЕ И ИХ СВОЙСТВА

В шестом классе вы научились вставлять в текстовый документ графические изображения. Это были рисунки, схемы, диаграммы и т. п. Их используют для иллюстрации содержания текстового документа. Вы уже знаете,

что графические объекты, вставленные в текстовый документ, имеют определенные свойства: *размер изображения, расположение на странице, обтекание текстом, контрастность и яркость, границы рисунка, заливка* и т. п.

В текстовый документ можно вставить различные графические изображения. Для этого используют элементы управления группы **Иллюстрации** вкладки **Вставка**:

- **Рисунок** – для вставки графических изображений, сохраненных на внешних носителях.
- **Клип** – для вставки графических изображений из готовых коллекций картинок, установленных на компьютере, или с сайта **Office Online** корпорации **Microsoft**.
- **Фигуры** – для вставки различных линий и геометрических фигур.
- **SmartArt** – для вставки разнообразных схем (организационных, иерархических, циклических и т. п.).
- **Диаграмма** – для вставки числовых диаграмм (гистограмм, секторных диаграмм, графиков и т. п.).

Из перечисленных видов вы уже умеете вставлять в текстовый документ графические изображения из коллекции **Microsof ClipArt**, из коллекции **SmartArt**, из различных коллекций графических изображений на внешних носителях, из Интернета, из собственных цифровых фотоальбомов и др.

В этом параграфе рассмотрим, как создать в текстовом документе графическое изображение, используя различные графические примитивы (рис. 3.20).

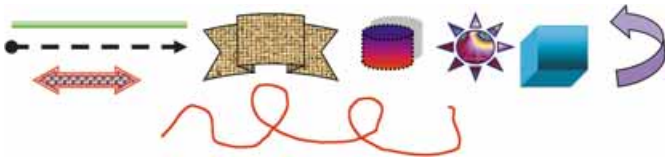


Рис. 3.20. Примеры графических примитивов с разными значениями их свойств

Графические примитивы есть двух видов – ***линия*** и ***фигура***.

Объект **линия** характеризуется следующими свойствами: *форма*, *толщина*, *тип*, *шаблон*, *цвет*. Возможные значения этих свойств приведены в таблице 3.7.

Таблица 3.7

Примеры значений свойств линии

Форма линии	*Толщина линии*	*Тип линии*	*Шаблон линии*	*Цвета линии*
Прямая Кривая Ломаная	0,25 пт 0,5 пт 0,75 пт 1 пт 1,5 пт 2,25 пт 3 пт 4,5 пт 6 пт			

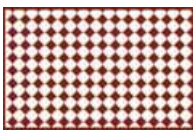
Фигуры образуются *замкнутыми линиями*. Эти замкнутые линии образуют **контур фигуры**, а часть плоскости, которую они ограничивают, – **внутреннюю область фигуры** (рис. 3.21). Внутренняя область фигуры может быть закрашена. При этом можно использовать один из 5 способов (эффектов) заливки: *однородную*, *градиентную*, *заливку узором*, *текстурой*, *рисунком* (табл. 3.8).



Рис. 3.21. Контур и внутренняя область фигуры

Таблица 3.8

Примеры закрашивания внутренней области фигуры

Однородная заливка	*Градиентная заливка*	*Заливка узором*	*Заливка текстурой*	*Заливка рисунком*
				
Одним цветом из палитры цветов	Оттенками нескольких цветов с плавными переходами между ними	Мозаичным рисунком, из коллекции шаблонов	Некоторым изображением из коллекции образцов	Изображением из любого графического файла

СОЗДАНИЕ ГРАФИЧЕСКИХ ПРИМИТИВОВ И ИЗОБРАЖЕНИЙ В WORD 2007



Рис. 3.22. Инструменты для создания графических примитивов

Графические примитивы, которые можно использовать в **Word 2007**, разделяют на несколько групп: *линии*, *основные фигуры*, *фигурные стрелки*, *элементы блок-схем*, *выноски*, *звезды и ленты* (рис. 3.22).

В **Word 2007** используется несколько видов линий – *прямая линия*, *соединительная линия*, *кривая*, *полилиния* и *рисованная кривая*. Для их создания используют следующие инструменты:

- **Прямая** \, **Стрелка** ↘, **Двусторонняя стрелка** ↖↘ – для рисования прямых линий. Создание этих объектов аналогично рисованию прямой линии в графическом редакторе **Paint**.
- **Рисованная кривая** – для создания произвольной кривой, имитирующей рисование линий карандашом от руки. Создание такого примитива аналогично рисованию карандашом в графическом редакторе **Paint**.
- **Кривая** – для создания кривой со сгибами.

- **Полилиния** – для создания ломаных линий.

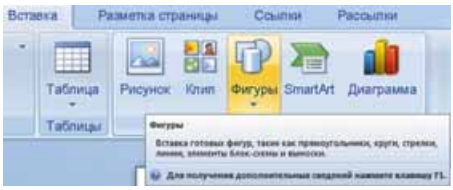


Рис. 3.23. Вставка фигуры

Алгоритм создания ломаной (аналогично создается и кривая):

1. Выбрать на вкладке **Вставка** в группе **Иллюстрации** кнопку **Фигуры** (рис. 3.23).
2. Выбрать инструмент **Полилиния**.
3. Выбрать на странице начальную точку линии.
4. Выбрать на странице точки, которые являются вершинами ломаной.
5. Дважды щелкнуть в конечной точке линии.

Для создания замкнутой ломаной (кривой) нужно ее конечную точку выбрать у начальной точки этой линии. Программа автоматически соединяет эти точки, замкнув линию и закрасив внутреннюю область.

Инструменты из групп **Основные фигуры**, **Фигурные стрелки**, **Блок-схемы** используются для создания примитивов, которые имеют заданную геометрическую форму. Например, ромб, треугольник, двойная стрелка, улыбающееся лицо, дуга, Луна и т. п. Рисование этих фигур выполняется по следующему алгоритму:

1. Выбрать на вкладке **Вставка** в группе **Иллюстрации** кнопку **Фигуры**.
2. Выбрать в приведенном списке нужную фигуру.
3. Перевести курсор в рабочую область документа.
4. Зафиксировать начальную точку рисования фигуры нажатием левой кнопки мыши.
5. Переместить указатель в конечную точку рисования фигуры, удерживая нажатой левую кнопку мыши.
6. Отпустить левую кнопку мыши.

Примечания:

- если, рисуя овал или прямоугольник, удерживать нажатой клавишу **Shift**, то будут создаваться фигуры правильной формы – круг или квадрат;
- содержание клавиши **Shift** при рисовании линий и стрелок ограничивает углы наклона величинами, кратными 15°;
- рисование фигур при нажатой клавише **Ctrl** осуществляется от центра фигуры;
- если нужно последовательно нарисовать несколько объектов одного вида, то соответствующий инструмент следует выбрать правой кнопкой и зафиксировать режим рисования. Добавив все необходимые объекты, нажать клавишу **Esc**;
- чтобы облегчить рисование мелких деталей изображения, целесообразно увеличить масштаб просмотра документа.

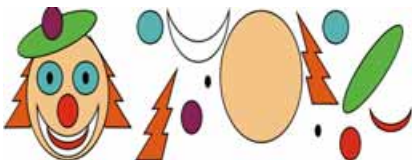


Рис. 3.24. Графическое изображение, созданное из отдельных графических примитивов

Каждый графический примитив в текстовом документе создается отдельно, а потом перетаскиванием, копированием, наложением друг на друга создается графическое изображение (рис. 3.24).

Для тех, кто хочет знать больше

При создании графического изображения в **Word 2007** удобно пользоваться сеткой, которая позволяет регулировать размеры графических объектов, упрощает рисование прямых линий и выравнивание объектов. Для отображения сетки нужно на вкладке **Вид** в группе **Показать или скрыть** установить флажок **Сетка** (рис. 3.25). Шаг сетки можно изменить в диалоговом окне **Привязка к сетке**, которое открывается ***Разметка страницы*** ⇒ ***Упорядочить*** ⇒ ***Выровнять*** ⇒ ***Параметры сетки***.

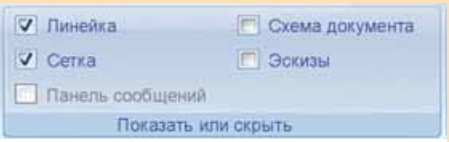


Рис. 3.25. Установка отображения сетки

Линии сетки отображаются на экране только в режиме просмотра **Разметка страницы** и не печатаются при выводе документа на печать. Для того чтобы скрыть сетку, нужно снять соответствующий флажок.

Для тех, кто работает с LibreOffice

Для создания графических объектов в текстовом документе используют команды меню **Вставка (Изображение, Медиа, Фигуры** и т. п.) или элементы управления панели инструментов **Графика** (рис. 3.26), которая появляется на экране при выборе кнопки **Функции рисования** на панели инструментов **Стандартная**. Кнопки **Основные фигуры**, **Фигуры-символы**, **Стрелки**, **Звезды**, **Выноски**, **Схемы** открывают списки эскизов соответствующей группы фигур.



Рис. 3.26. Панель инструментов **Графика**

РЕДАКТИРОВАНИЕ ГРАФИЧЕСКИХ ОБЪЕКТОВ В WORD 2007

Графические примитивы являются самостоятельными объектами и их можно редактировать независимо от других объектов этого изображения. Напоминаем, что все эти операции осуществляются для выделенных объектов. Чтобы выделить объект, нужно выбрать его на изображении. При наведении указателя на объект он меняет вид на такой: ✥.

После выбора объекта вокруг него появляются *маркеры изменения размеров*, а для некоторых графических объектов также и *маркер вращения* – это специальные отметки в виде маленьких кружков и квадратиков, расположенных вокруг фигуры (рис. 3.27).

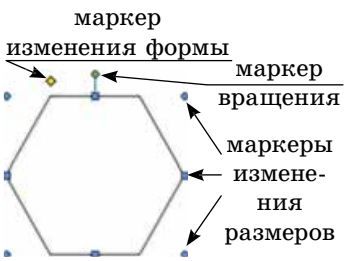


Рис. 3.27. Маркеры графического объекта

При наведении указателя на маркеры изменения размеров он приобретает вид двусторонней стрелки, указывающей направление возможного перемещения. Перемещение маркера в этих направлениях приводит к соответствующему изменению размеров фигуры.

Используя маркер вращения, объект можно повернуть на произвольный угол. Для этого следует навести указатель на маркер вращения (указатель изменит свой вид на такой:), затем перемещать мышь по кругу в нужную сторону, удерживая нажатой левую кнопку мыши.

Для некоторых фигур существует еще один маркер – *маркер изменения формы*, который имеет вид маленького желтого ромбика. Используя его, можно менять внешний вид фигур, при этом их ширина и высота будут оставаться неизменными. Например, можно изменить вид острия стрелки, как это показано на рисунке 3.28. Все эти фигуры получены модификацией формы первой фигуры.



Рис. 3.28. Модификация фигуры маркером изменения формы

Для тех, кто хочет знать больше

Изменение формы линий, в том числе кривых, осуществляется редактированием их узлов (рис. 3.29). Для этого следует:

1. Выделить линию.
2. Выбрать кнопку **Изменить фигуру** , которая размещается на вкладке **Формат** в группе **Вставить фигуры**.
3. Выбрать команду **Начать изменение узлов**.
4. Перетащить узлы линии (маленькие черные квадратики) в нужное место.

Операции копирования, перемещения, удаления графических объектов осуществляются известными вам способами: с использованием **Буфера обмена**, перетаскиванием или клавишами клавиатуры.

Если графический объект нужно переместить на незначительное расстояние, то для выделенного объекта это можно сделать нажатием клавиш управления курсором в соответствующем направлении. Шаг такого перемещения будет равен установленному по умолчанию значению. Для его уменьшения при перемещении следует дополнительно удерживать нажатой клавишу **Ctrl**.

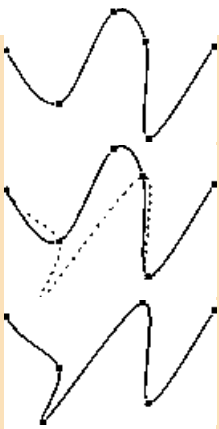


Рис. 3.29. Модификация кривой с использованием узлов

ФОРМАТИРОВАНИЕ ГРАФИЧЕСКИХ ОБЪЕКТОВ В WORD 2007

У созданного графического объекта можно изменять значения его свойств, то есть форматировать. К операциям форматирования графических примитивов относятся: *изменение формы и размеров объекта, цвета, типа и шаблона линий его контура, цвета и эффектов заливки* и т. п.

Для форматирования графических примитивов удобно пользоваться элементами управления временной вкладки **Формат** (рис. 3.30), которая появляется на **Ленте** в разделе **Средства рисования** при выделении графического объекта.

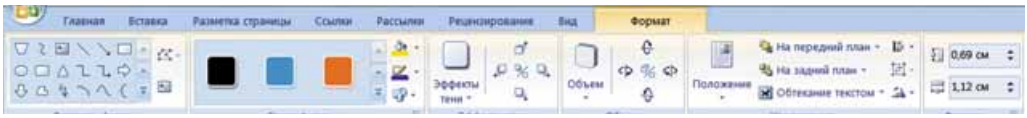


Рис. 3.30. Вкладка **Формат**

Назначение этих элементов управления приведено в таблице 3.9. Во время выполнения некоторых операций форматирования применяется динамический просмотр результатов форматирования.

Таблица 3.9

Назначение элементов управления вкладки Формат

Элемент управления	*Название*	*Назначение*
Группа **Стили фигур**		
	Список **Экспресс-стили**	Для выбора стиля оформления объекта из предоставленного списка образцов
	Кнопка со списком **Заливка фигуры**	Для заливки внутренней области объекта установленным цветом. Выбор кнопки открытия списка открывает панель для выбора цвета и эффекта заливки
	Кнопка со списком **Контур фигуры**	Для изменения цвета контура фигуры на установленный цвет. Выбор кнопки открытия списка открывает панель для выбора цвета линии, ее толщины, типа и т. п.
	Кнопка со списком **Изменить фигуру**	Для замены выделенной фигуры на другую с сохранением установленного форматирования
Группа **Эффекты**		
	Кнопка **Эффекты тени**	Для установки значений свойств тени объекта
	Кнопка **Объем**	Для установки значений свойств объемного изображения объекта
Группа **Размер**		
	Счетчик **Высота фигуры**	Для установки значения высоты объекта
	Счетчик **Ширина фигуры**	Для установки значения ширины объекта

Установление значений свойств графических объектов можно выполнять и в диалоговом окне **Формат автофигуры**, которое открывается одноименной командой контекстного меню этого объекта. После этого следует

установить нужные значения на вкладках **Цвет и линии**, **Размер** и **Положение**.

Для графических объектов, как и для текстовых фрагментов, можно применять форматирование по образцу, которое позволяет отформатировать графический объект так же, как и какой-либо другой.

Для тех, кто работает с LibreOffice

Для установки значений свойств графических объектов используются элементы управления панели инструментов **Свойства рисунка** (рис. 3.31).



Рис. 3.31. Панель инструментов **Свойства рисунка**

ОПЕРАЦИИ НАД ГРУППОЙ ГРАФИЧЕСКИХ ОБЪЕКТОВ

Если некоторые операции редактирования или форматирования нужно выполнить не с одним объектом, а с несколькими, то для ускорения работы следует выделить группу этих объектов и выполнить нужные операции для всех объектов группы одновременно.

Выделение группы графических объектов в **Word 2007** происходит при нажатой клавише **Ctrl** с использованием мыши, аналогично выполнению подобной операции с объектами файловой системы: папками, файлами, ярлыками. Кроме того, для выделения группы графических объектов в некоторой прямоугольной области рисунка используется кнопка **Выбор объектов**, которая расположена на вкладке **Главная** в группе **Редактирование** в меню команды **Выделить**. Выбрав эту кнопку, нужно выделить прямоугольный участок документа – и все графические объекты, которые полностью попали в эту область, будут выделены.

Примечание: при выборе этой кнопки включается режим выделения графических объектов, и обработка текстовых объектов документа в этом режиме невозможна. Отменить этот режим можно повторным выбором этой кнопки.

Для того чтобы группа выделенных объектов воспринималась как единое целое, эти объекты нужно *сгруппировать*. В дальнейшем все операции редактирования или форматирования будут применяться сразу ко всем объектам группы. Такое группирование проводится, когда создание рисунка завершено.

Выполнение операции группировки осуществляется следующим образом:

1. Выделить нужные графические объекты изображения.
2. Выбрать на вкладке **Формат** в группе **Упорядочить** кнопку **Группировать**.
3. Выбрать команду **Группировать**.

После группировки сохраняется возможность выделить любой отдельный объект и выполнить над ним некоторые операции.

При необходимости объекты можно *разгруппировать* или *перегруппировать*, используя соответствующие команды кнопки **Группировать**.

Операции группировки и разгруппировки объектов можно осуществлять и с использованием команды **Группировка** контекстного меню объектов (рис. 3.32).

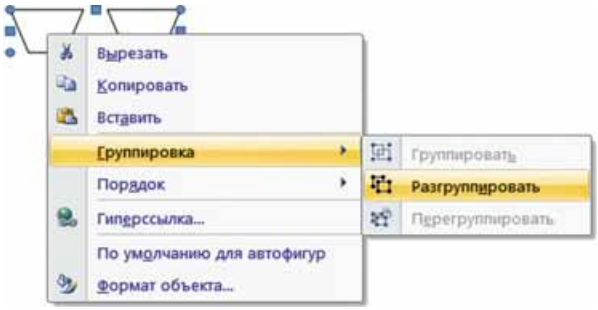


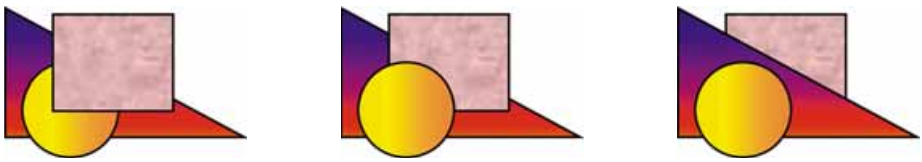
Рис. 3.32. Операции команды **Группировка контекстного меню**

Для тех, кто хочет знать больше

Упорядочение объектов графического изображения

Изображение, созданное из отдельных графических объектов, имеет *многослойную структуру* – отдельные объекты могут быть расположены поверх других. Порядок их наложения друг на друга можно менять (рис. 3.33), перемещая объект относительно других объектов в верхние или нижние слои.

Над объектами или группами объектов изображения можно выполнять и другие операции – изменять *порядок их расположения* на картинке, *упорядочивать* объекты на странице, *поворачивать* и *отражать* и т. п.



Первоначальное изображение группы объектов | Перемещение квадрата на слой ниже | Перемещение квадрата на задний план

Рис. 3.33. Пример перемещения квадрата на нижние слои изображения

Созданные графические объекты или группу объектов можно повернуть направо или налево на 90°, отразить сверху вниз или слева направо (рис. 3.34).



Рис. 3.34. Вращение группы графических объектов

Выполнение всех этих операций с группой графических объектов осуществляется с использованием элементов управления группы **Упорядочить** временной вкладки **Формат** (табл. 3.10) или соответствующими командами контекстного меню графических объектов.

Таблица 3.10

Назначение элементов управления группы Упорядочить вкладки Формат

Элемент управления	*Название*	*Назначение*
	Кнопка со списком **Положение в тексте**	Для выбора размещения объекта на странице документа
	Кнопка со списком **На передний план**	Для перемещения объекта в верхний слой многослойного изображения
	Кнопка со списком **На задний план**	Для перемещения объекта в нижний слой многослойного изображения
	Кнопка со списком **Обтекание текстом**	Для выбора способа обтекания объекта текстом
	Кнопка со списком **Выровнять**	Для выбора способа взаимного расположения объектов группы на странице
	Кнопка со списком **Группировать**	Для выбора операции над группой объектов
	Кнопка со списком **Повернуть**	Для выбора операций поворота и отражения объекта

Самое важное в этом пункте

В текстовый документ можно вставить различные графические изображения – рисунки, фотографии, схемы и т. п. Одна из разновидностей таких изображений это графические примитивы – линии и фигуры.

Объект **линия** характеризуется следующими свойствами: *форма, толщина, тип, шаблон, цвет*. Фигуры образуются замкнутыми линиями. Эти замкнутые линии образуют *контур фигуры*, а часть плоскости, которую они ограничивают, – *внутреннюю область фигуры*. Внутренняя область фигуры может быть закрашена.

Графические примитивы, которые можно использовать в **Word 2007**, разделяются на несколько групп: Линии, Основные фигуры, Фигурные стрелки, Элементы блок-схем, Выноски, Звезды и ленты. Инструменты для вставки всех этих графических объектов находятся в списке кнопки **Фигуры** на вкладке **Вставка** в группе **Иллюстрации.**

Каждый графический примитив в текстовом документе создается отдельно, а затем перетаскиванием, копированием, наложением друг на друга создается графическое изображение.

Для выполнения операций над объектом его нужно выделить. После выделения объекта вокруг него появляются маркеры изменения размеров, маркер вращения, а для некоторых фигур – и маркер изменения формы, которые позволяют редактировать созданный объект. Операции копирования, перемещения, удаления графических объектов осуществляются с использованием **Буфера обмена**, перетаскиванием или клавишами клавиатуры.

Созданный графический примитив можно форматировать: менять цвет, тип и шаблон линий его контура, цвет и эффекты заливки и т. п. Для форматирования графических примитивов используются элементы управления временной вкладки **Формат**, которая появляется на **Ленте** в разделе **Средства рисования** при выделении графического объекта.

Дайте ответы на вопросы

1°. Из каких графических примитивов может состоять графическое изображение в **Word 2007**?
2•. Какие свойства имеет линия? Какие значения они могут приобретать?
3•. Какие свойства имеет фигура? Какие значения они могут приобретать?
4•. Линии каких видов можно построить в **Word 2007**? Какие инструменты для этого используют?
5•. Для чего используют клавиши **Ctrl** и **Shift** при рисовании графических примитивов?
6°. Для чего предназначены маркеры изменения размера, маркер вращения? Как ими пользоваться?
7°. Какие эффекты заливки можно использовать в текстовом процессоре **Word 2007**?
8°. Как в текстовом документе выделить графический объект или группу объектов?
9•. Какие операции редактирования и форматирования можно выполнять с графическими примитивами в текстовом документе **Word 2007**?
10•. С какой целью выполняют группировку объектов? Как это сделать?

Выполните задания

1•. Нарисуйте ломаную линию по алгоритму, приведенному в учебнике. Скопируйте ее несколько раз. Отформатируйте объекты по-разному. Сохраните документ в вашей папке с именем **задание 3.3.1.docx**.
2•. Нарисуйте звезду. Скопируйте ее 5 раз и разместите звезды по кругу. Сохраните документ в вашей папке с именем **задание 3.3.2.docx**.
3•. Нарисуйте трапецию. Скопируйте ее несколько раз. Для каждого объекта примените различные виды теней, объемных эффектов, эффектов заливки. Сохраните документ в вашей папке с именем **задание 3.3.3.docx**.
4°. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.3.\образец 3.3.4.docx**). Переместите объекты на изображении в порядке 3-1-2. Измените размеры объектов изображения: маленькие объекты увеличьте, а большие – уменьшите. Сохраните документ в вашей папке с именем **задание 3.3.4.docx**.
5°. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.3.\образец 3.3.5.docx**). Удалите с изображения все стрелки. Измените форму каждого объекта с помощью маркера изменения формы. Сохраните документ в вашей папке с именем **задание 3.3.5.docx**.
6°. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.3.\образец 3.3.6.docx**). Скопируйте изображенный объект трижды. По-

верните четыре изображения в разные стороны. Сохраните документ в вашей папке с именем **задание 3.3.6.docx.**

7°. Нарисуйте круг и звезду, расположите их рядом. Сгруппируйте их в единый объект. Сохраните документ в вашей папке с именем **задание 3.3.7.docx.**

8*. Найдите на вашем компьютере графический файл формата **WMF.** Вставьте его в текстовый документ. Разгруппируйте его элементы и измените отдельные части рисунка (размеры, цвета, заливки и т. п.). Сохраните документ в вашей папке с именем **задание 3.3.8.docx.**

3.4. СОЗДАНИЕ, РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ СПЕЦИАЛЬНЫХ ГРАФИЧЕСКИХ ОБЪЕКТОВ

1. Как можно ввести в текст математические символы, например x_3, x^3, $^3/_8$, α?
2. Какие графические изображения можно вставлять в текстовый документ?
3. Какие операции редактирования и форматирования можно осуществлять над графическими объектами в текстовом процессоре **Word 2007**? С использованием каких средств осуществляются эти операции?

СОЗДАНИЕ ФОРМУЛ В ТЕКСТОВОМ ДОКУМЕНТЕ

При подготовке в текстовом процессоре реферата для урока или научной статьи для ученической конференции в документ иногда нужно вставлять формулы, уравнения, доказательства теорем, которые используют особую символику: математическую, физическую, химическую и др.

Несложные формулы и выражения имеют линейный вид, то есть все элементы записаны в одной строке, в них нет дробей, знаков корней и других нелинейных структур. Например:

а) $c_2m_2(t_2 - \theta) = c_1m_1(\theta - t_1)$;

б) $ax^2 + bx + c \geqslant 0$;

в) $(x^3 + 2x^2y - 5xy^2 - 3y^3)(5x - 4y)$;

г) $Na_2CO_3 + 2HCl \Leftrightarrow 2NaCl + CO_2 + H_2O$.

Для создания таких выражений используют разные специальные символы, вставка которых осуществляется в окне **Символ** (***Вставка*** ⇒ ***Символы*** ⇒ ***Символ*** Ω ⇒ ***Другие символы***). Ввод символов нижнего и верхнего индекса осуществляется с использованием кнопок **Подстрочный знак** x_2 и **Надстрочный знак** x^2, которые расположены на вкладке **Главная** в группе **Шрифт.**

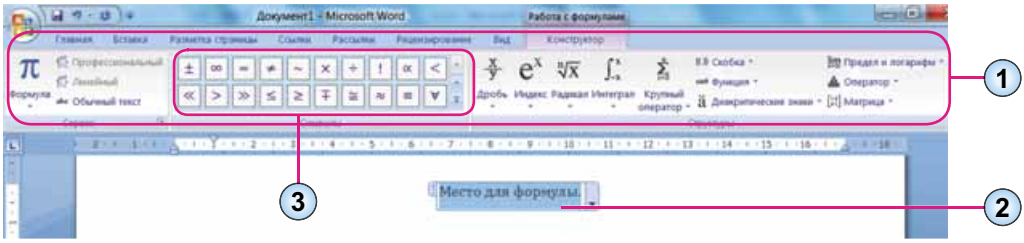
Однако часто встречаются нелинейные, многоуровневые формулы и выражения. Например:

а) $\frac{h_1}{h_2} = \frac{\rho_2}{\rho_1}$; б) $E_к = \frac{mv^2}{2}$; в) $\eta = \frac{A_1}{A_2} \cdot 100\,\%$; г) $\frac{3x+7}{4} - \frac{x-3}{2} = \frac{5x+2}{8}$;

д) $y = \begin{cases} -3x^3 - 2, & \text{при } x < 0, \\ \frac{7x}{4} - 2, & \text{при } x \geqslant 0; \end{cases}$ е) $\frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a-4b)^2}{9a^4 - b^4}$.

Вставить в текстовый документ такие формулы можно, воспользовавшись специальным средством **Конструктор формул.** Формулы в нем конструируются из отдельных символов и структур, используя соответствующие шаблоны.

Для открытия **Конструктора формул** нужно выполнить ***Вставка*** ⇒ ***Символы*** ⇒ ***Формула*** π. После этого на **Ленте** в разделе **Работа с формулами** появится временная вкладка **Конструктор** (рис. 3.35, 1), а в документе – специальная область для ввода формулы (рис. 3.35, 2).



1. Вкладка **Конструктор**
2. Место для ввода формулы
3. Кнопки для вставки символов

Рис. 3.35. Вкладка **Конструктор**

Для записи формулы некоторые буквы, знаки арифметических операций и другие символы можно вводить с клавиатуры, а специальные – выбирать в соответствующих списках группы **Символы.**

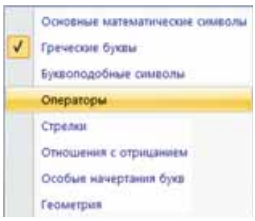


Рис. 3.36. Список основных наборов символов

На **Ленте** отображаются только символы того набора, который использовался последним (рис. 3.35, 3). Для просмотра других частей списка символов текущего набора следует воспользоваться кнопками ▲ и ▼ на полосе прокрутки или кнопкой **Дополнительные параметры** ▼ для открытия всего списка символов этого набора. Чтобы открыть перечень всех наборов символов, нужно в текущем списке выбрать его название.

Все символы, которые можно вставить в формулу, разделены на 8 основных наборов (рис. 3.36), которые в свою очередь могут быть разделены на отдельные блоки (рис. 3.37).

Просмотрев списки и найдя нужный символ, его следует выбрать и вставить в текущее место выражения.



Рис. 3.37. Открытый список символов

Кроме отдельных символов, в выражении могут быть различные структуры – обыкновенные дроби, функции, скобки, знаки корней и т. п. Они вставляются в выражение с помощью элементов управления группы **Структуры** вкладки **Конструктор.**

Все структуры разделены по типам, список шаблонов которых открывается при выборе соответствующих кнопок на вкладке (рис. 3.38). Для того чтобы вставить

шаблон в выражение, следует открыть список шаблонов соответствующей кнопки и выбрать нужный.

Далее следует заполнить шаблон данными. Места для ввода цифр и символов в шаблоне обозначены специальными полями, обведенными пунктирной рамкой (рис. 3.39).

В поля можно вводить не только отдельные символы, но и вставлять другие шаблоны.



Рис. 3.38. Группа **Структуры** и список шаблонов кнопки **Корни**

Перемещение в пределах выражения осуществляется клавишами управления курсором или указателем мыши. Для выхода из режима создания формул нужно выбрать область за ее пределами.

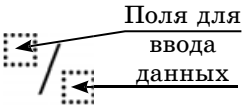


Рис. 3.39. Шаблон обыкновенной дроби

Редактирование созданной формулы или ее фрагментов (удаление, вставка или замена, перемещение и копирование) осуществляется стандартными средствами **Word 2007**. При форматировании можно изменить значения свойств отдельных символов формулы, установить междустрочный интервал, отступы, выравнивание, способ расположения формулы в документе и т. п. Для этого используются элементы управления **мини-панели** вкладки **Главная** и команды контекстного меню (рис. 3.40) формулы.

Созданную формулу пользователь может добавить в библиотеку встроенных формул. Для этого следует:

1. Выделить созданную формулу.
2. Выполнить ***Конструктор*** ⇒ ***Сервис*** ⇒ ***Формула.***
3. Выбрать в списке команду **Сохранить выделенный фрагмент в коллекцию формул.**
4. Ввести в диалоговом окне **Создание нового стандартного блока** имя созданной формулы, указать значения других параметров.
5. Выбрать кнопку **ОК**.

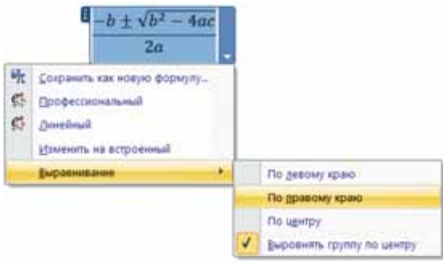


Рис. 3.40. Контекстное меню формулы

Для тех, кто работает с LibreOffice

Для вставки формулы в текст документа следует выполнить ***Вставка*** ⇒ ***Объект*** ⇒ ***Формула*** (рис. 3.41). В дальнейшем шаблоны используются для блоков формул, перечень которых появляется на панели в левой части окна программы. Редактирование формулы осуществляется в линейном виде в нижнем поле окна программы, а в тексте отображается результат редактирования.

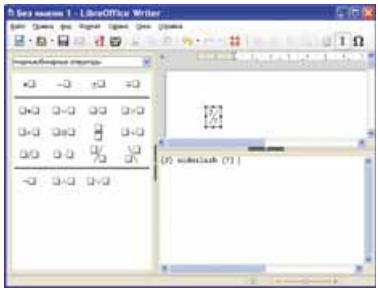


Рис. 3.41. Вставка формул в **LibreOffice**

Для тех, кто хочет знать больше

Создание надписей в текстовом документе

Для размещения надписей в текстовом документе существует несколько способов – текст можно разместить внутри фигуры или создать специальные графические объекты **Надпись** или **WordArt**.

Во внутренней области большинства фигур можно разместить текст. Для добавления текста к фигуре следует в контекстном меню фигуры выбрать команду **Прибавить текст** и ввести его. Если повернуть или зеркально отобразить фигуру, размещение текста внутри фигуры будет соответственно меняться (рис. 3.42).

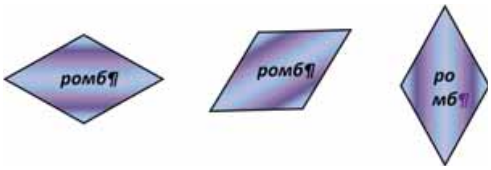


Рис. 3.42. Размещение текста при изменении фигуры



Рис. 3.43. Направление размещения текста внутри фигуры

Текст, размещенный в фигуре, можно редактировать и форматировать стандартными средствами **Word 2007.** Направление расположения текста внутри фигуры (горизонтально или вертикально) можно изменить последовательным выбором кнопки **Направление текста**, которая расположена на вкладке **Формат** в группе **Текст**. Варианты размещения текста представлены на рисунке 3.43.

Если нужно создать текстовую надпись как отдельный графический объект в текстовом документе, то для этого следует воспользоваться специальным инструментом **Надпись**, который размещен на вкладке **Вставка** в группе **Иллюстрации** в списке кнопки **Фигуры.**

Редактирование и форматирование текста в надписи осуществляется стандартными способами. В дальнейшем все операции с объектом **Надпись** (изменение размеров рамки, ее размещение и оформление) выполняются таким же образом, как и с другими графическими объектами.

Текстовый процессор **Word** позволяет вставлять в документы так называемые объекты **WordArt** (рис. 3.44), которые используются для художественного оформления документа.



Рис. 3.44. Пример объекта **WordArt**

Объекты **WordArt** имеют два вида свойств: *текстовые* (шрифт, размер символов, цвет, начертание) и *графические* (размер изображения, расположение, заливка, цвет линий, наличие тени и объемных эффектов и т. п.).

Создание объектов **WordArt** выполняется по следующему алгоритму:

1. Выбрать на вкладке **Вставка** в группе **Текст** кнопку **WordArt**.
2. Выбрать в приведенном списке нужный стиль надписи.
3. Ввести текст создаваемой надписи в текстовое поле.
4. Выбрать точку за пределами поля.

Созданный объект **WordArt** будет размещен в текущем месте документа. Редактирование и форматирование текста и графического объекта осуществляется с использованием элементов управления временной вкладки **Формат**.

Самое важное в этом пункте

Для создания несложных формул, которые имеют линейный вид, используют специальные символы, вставка которых осуществляется в окне **Символ (*Вставка* ⇒ *Символы* ⇒ *Символ* Ω ⇒ *Другие символы*).** Ввод символов нижнего и верхнего индекса осуществляется с использованием кнопок **Подстрочный знак** x_2 и **Надстрочный знак** x^2, которые расположены на вкладке **Главная** в группе **Шрифт.**

Введение многоуровневых формул осуществляется специальным средством **Конструктор формул (*Вставка* ⇒ *Сервис* ⇒ *Формула*).** Формулы в нем конструируются из отдельных символов и структур, используя соответствующие шаблоны, которые нужно заполнить данными.

Дайте ответы на вопросы

1°. Как ввести текст, в котором есть химические формулы?
2°. Как вставить в текст следующие математические знаки: ± ⩾ ≠ ≈?
3•. Какой инструмент можно использовать для создания формул в текстовом документе?
4°. Какие наборы математических символов используются в формулах? Как выбрать нужный символ?
5•. Какие структуры можно вводить в формулы? Назовите некоторые из них.
6•. Что такое шаблон структуры? Из каких элементов он состоит? Как его используют?
7•. Какие операции редактирования можно выполнять над формулой? Как они осуществляются?
8•. Какие операции форматирования можно выполнять над формулой? Как они осуществляются?

Выполните задания

1•. Создайте новый текстовый документ. Введите такой текст и сохраните документ в вашей папке в файле с именем **задание 3.4.1.docx.** В состав напитка «Кока-кола» входят: вода H_2O, углекислый газ CO_2, угольная H_2CO_3 и фосфорная H_3PO_4 кислоты, сахароза $C_{12}H_{22}O_{11}$, кофеин $C_8H_{10}N_4O_2$ и т. п.

2•. Создайте новый текстовый документ. Введите указанные схемы химических реакций. Сохраните документ в вашей папке в файле с именем **задание 3.4.2.docx.**

а) $O_2 + 2H_2 \rightarrow 2H_2O$;

б) $AgCl \rightarrow Ag + Cl_2$;

в) $K + H_2O \rightarrow KOH + H_2\uparrow$;

г) $KMnO_4 \rightarrow K_2MnO_4 + MnO_2 + O_2\uparrow$.

3•. Создайте новый текстовый документ. Вставьте в него указанные физические формулы. Сохраните документ в вашей папке в файле с именем **задание 3.4.3.docx.**

а) период вращения $T = \frac{t}{N}$;

б) формула сообщающихся сосудов $\frac{h_1}{h_2} = \frac{\rho_2}{\rho_1}$;

в) кинетическая энергия $E_к = \frac{mv^2}{2}$;

г) коэффициент полезного действия $\eta = \frac{A_1}{A_2} \cdot 100\,\%$.

4•. Создайте новый текстовый документ. Вставьте в него указанные математические выражения. Сохраните документ в вашей папке в файле с именем **задание 3.4.4.docx**.

а) $\frac{3}{4}(12 - x) = \frac{3}{8}$;

в) $\frac{7}{8}(2 - x) = \frac{1}{8}$;

б) $\frac{3}{7}(1 - 2x) + \frac{1}{7} = -2$;

г) $\frac{2}{3}(x - 4) + \frac{1}{3} = 5$.

5•. Создайте новый текстовый документ. Вставьте в него указанные математические выражения. Сохраните документ в вашей папке в файле с именем **задание 3.4.5.docx**.

а) $y = \begin{cases} x^2 + 3, \text{ при } x \leqslant 0, \\ \frac{x}{2} + 3, \text{ при } x > 0; \end{cases}$

в) $y = \begin{cases} 3x - y = 1, \\ x + y = 3, \\ 4y = 8; \end{cases}$

б) $y = \begin{cases} -3x^3 - 2, \text{ при } x < 0, \\ \frac{7x}{4} - 2, \text{ при } x \geqslant 0; \end{cases}$

г) $y = \begin{cases} x - y = 1, \\ x + y = 3, \\ 2x = 6. \end{cases}$

6•. Создайте новый текстовый документ. Вставьте в него указанные математические выражения. Сохраните документ в вашей папке в файле с именем **задание 3.4.6.docx**.

а) $\frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a - 4b)^2}{9a^4 - b^4}$;

в) $\frac{a + \frac{25}{a + 10}}{\frac{25}{a} - a}$;

б) $\frac{x}{7a^2b^3} \cdot \frac{6a^3c^2}{b} = \frac{y}{4c}$;

г) $1 - \frac{1}{1 - \frac{a}{1 - \frac{1}{a + 1}}}$.

ПРАКТИЧЕСКАЯ РАБОТА № 4

«Создание текстового документа, содержащего объекты разных типов»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте текстовый документ из файла **образец пр4.docx**, который находится в папке **Глава 3\практическая 4**. Оформите его по указанному образцу:

Дорогие дети!

Юношеская туристическая фирма **«Топ-топ»** предлагает провести незабываемые веселые каникулы на легендарном острове Хортица!

Вас ожидают:

- путешествия,
- экскурсии,
- дискотеки,
- туристические эстафеты,
- спортивные конкурсы,
- казацкие забавы,
- прогулки по реке,
- удивительные скалы,
- песчаные пляжи,
- исторические места.

Для проживания можно выбрать:

1. Лагерь пластунов *(7 дней, до 20 детей)*
2. Стационарные корпуса *(20 дней, до 500 детей)*
3. Деревянные домики *(10 дней, до 100 детей)*

Стоимость путевок:

Срок	Цена (за 1 место в день) в грн		
			
Март - май	20	50	70
Июнь - август	30	70	100
Сентябрь - октябрь	20	50	70

Заказать путевки можно:
на сайте **www.top-top.zp.ua** или
по электронной почте **admin@top-top.zp.ua**

2. Создайте в документе логотип туристической фирмы «Топ-топ» из графических примитивов согласно образцу:



3. Введите в конце документа указанные формулы:

а) $\rho = \rho_0 \dfrac{P}{P - P_1}$;

б) $KMnO_4 \rightarrow K_2MnO_4 + MnO_2 + O_2\uparrow$;

в) $\dfrac{3x+7}{4} - \dfrac{x-3}{2} = \dfrac{5x+2}{8}$.

4. Сохраните документ в вашей папке в файле с именем **практическая работа 4.docx**.

3.5. РАЗДЕЛЫ, КОЛОНТИТУЛЫ, СТИЛИ, СТРУКТУРА В ТЕКСТОВЫХ ДОКУМЕНТАХ

1. Какие свойства страниц вы знаете? Как установить значения этих свойств?
2. Какие режимы просмотра текстового документа существуют в **Word 2007**? Как их установить?
3. Каким образом структурирован текст вашего учебника по информатике? Какие объекты находятся на его страницах?

СТРУКТУРА ДОКУМЕНТА

Многостраничные документы (рефераты, курсовые работы, брошюры, книги и т. п.) удобно делить на структурные части – разделы, главы, параграфы, пункты и т. п., создавая таким образом иерархическую структуру документа. С такими структурами документов вы сталкивались, например, на уроках украинской литературы, когда составляли план написания сочинения.

Если в качестве примера рассмотреть структуру этого учебника (рис. 3.45), то на первом уровне иерархии находятся разделы учебника, второй уровень составляют пункты, третий – подпункты и т. п.

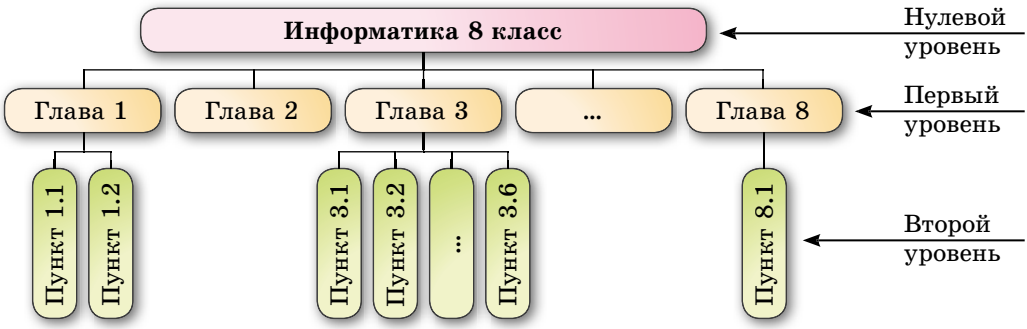


Рис. 3.45. Пример структуры документа

Структура документа – это иерархическая схема размещения составных частей документа.

РАЗДЕЛЫ В ТЕКСТОВОМ ДОКУМЕНТЕ

Для того чтобы разделить текстовый документ на разделы в том месте, где будет начинаться другой раздел, нужно вставить **Разрыв раздела**. Если разделов в документе должно быть несколько, то для каждого из разделов вставляется свой разрыв. При отсутствии разрывов разделов текстовый документ рассматривается как один раздел.

Существует несколько видов разрывов разделов:

- со следующей страницы – раздел начинается с начала следующей страницы;
- с текущей страницы – раздел начинается в указанном месте документа;
- с четной страницы – раздел начинается со следующей четной страницы;
- из нечетной страницы – раздел начинается со следующей нечетной страницы.

Создание разделов желательно выполнять в режиме просмотра документа **Разметка страницы** и при включенном режиме отображения *скрытых* (*непечатаемых*) символов. Далее нужно выполнить ***Разметка страницы*** ⇒ ***Параметры страницы*** ⇒ ***Разрывы*** и выбрать нужный тип в списке **Разрывы разделов**. В месте разрыва разделов появится скрытый (непечатаемый) символ **Разрыв раздела**.

Например, такой: ¶ Разрыв раздела (со следующей страницы)

Обычно в текстовом документе имеется только один раздел, и все страницы документа отформатированы одинаково (размер страницы, размеры полей, ориентация страницы и т. п.). Однако при работе с многостраничными документами может потребоваться, чтобы некоторые страницы документа имели другие параметры. Например, в тексте с книжной ориентацией страниц нужно вставить таблицу или схему на странице с альбомной ориентацией или какой-то фрагмент документа нужно разместить в несколько колонок. Такое форматирование документа возможно сделать, разделив документ на разделы.

Для удаления разрыва разделов нужно удалить в документе соответствующий скрытый символ **Разрыв раздела**. При этом текст над местом удаленного разрыва становится частью следующего раздела и его формат автоматически изменяется соответствующим образом.

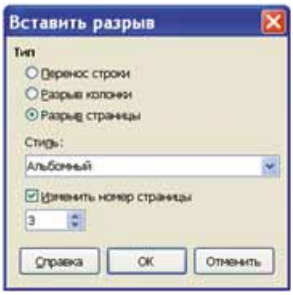


Рис. 3.46. Окно **Вставить разрыв**

Для тех, кто работает с LibreOffice

Для создания разрыва в тексте нужно выполнить ***Вставка*** ⇒ ***Разрыв*** и выбрать в списке нужный тип разрыва (рис. 3.46). Вставить разрыв можно также выбором кнопки **Вставить разрыв страницы** на панели инструментов **Стандартная**.

ВСТАВКА КОЛОНТИТУЛОВ

В многостраничных документах часто используют **колонтитулы** (франц. ***colonne*** – столбец, колонка, лат. ***titulus*** – заголовок) – **это служебные сообщения, которые размещаются на полях страницы и повторяются на других страницах документа.**

В **Word 2007** различают *верхний, нижний* и *боковые* колонтитулы. Они могут содержать номера страниц, название документа или текущего раздела, фамилию автора, графические изображения и т. п. Для примера – рассмотрите страницу вашего учебника по информатике.

Колонтитулы первой страницы, четных и нечетных страниц могут отличаться. Также могут различаться колонтитулы разных разделов документа, например, как в этом учебнике – в разных разделах разные колонтитулы.

В текстовый документ в **Word 2007** можно вставлять колонтитулы из готовой коллекции шаблонов колонтитулов или создавать собственные колонтитулы, которые можно сохранить в коллекции. В документе колонтитулы отображаются только в режимах **Разметка страницы** и **чтение.** Обработка текста документа при работе с колонтитулами невозможна.

Для создания колонтитулов используются элементы управления вкладки **Вставка** группы **Колонтитулы** (рис. 3.47).



Рис. 3.47. Группа **Колонтитулы**

После их создания на **Ленте** появляется временная вкладка **Конструктор** (рис. 3.48). Используя соответствующие кнопки групп **Колонтитулы** и **Вставить**, в колонтитул можно вставить номер страницы – кнопка **Номер страницы**, текущую дату и время – кнопка **Дата и время**, рисунок – кнопка **Рисунок** и др. Размещение текста колонтитула относительно верхнего или нижнего края страницы регулируется соответствующими счетчиками в группе **Положение**.

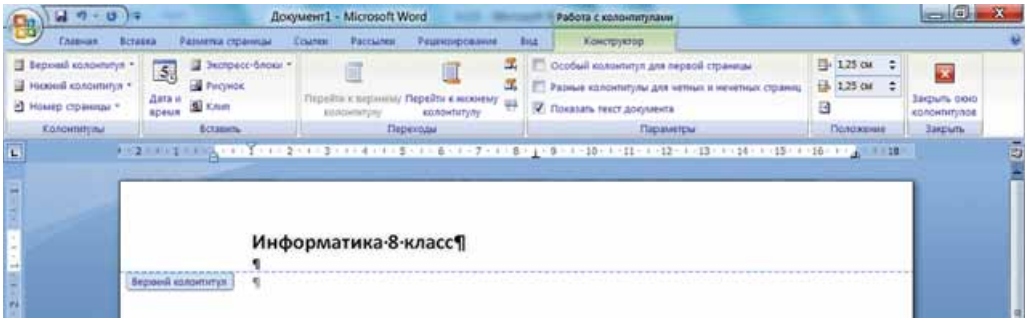


Рис. 3.48. Шаблон верхнего колонтитула и временная вкладка **Конструктор**

Для создания на каждой странице документа колонтитула из готовой коллекции шаблонов нужно:

1. Выбрать на **Ленте** вкладку **Вставка**.
2. Выбрать в группе **Колонтитулы** одну из команд **Верхний колонтитул** или **Нижний колонтитул**.
3. Выбрать в предложенном списке шаблонов колонтитулов один из вариантов.
4. Установить нужные значения свойств колонтитула, используя элементы управления вкладки **Конструктор**.
5. Ввести текст в поля шаблона.
6. Закрыть колонтитул, выполнив ***Работа с колонтитулами*** ⇒ ***Конструктор*** ⇒ ***Закрыть*** ⇒ ***Закрыть окно колонтитулов***.

Созданный колонтитул можно применить или ко всему документу, или к четным (нечетным) страницам, или к первой странице. Выбрать область применения созданного колонтитула можно в списке кнопки **Параметры**. Установить значения некоторых свойств колонтитулов можно также и на вкладке **Источник бумаги** диалогового окна **Параметры страницы** вкладки **Разметка страницы**.

Для перехода с поля верхнего колонтитула в поле нижнего колонтитула и обратно используется кнопка **Перейти к верхнему (нижнему) колонтитулу** в группе **Переходы** вкладки **Конструктор**.

Если колонтитулы для разных частей документа должны быть разными, то перед их созданием документ нужно разделить на разделы. Затем колонтитулы для разных разделов создаются отдельно.

При необходимости текст колонтитула можно редактировать и форматировать обычным способом.

Для удаления колонтитула следует выполнить ***Вставка*** ⇒ ***Колонтитулы*** ⇒ ***Верхний колонтитул (Нижний колонтитул)*** ⇒ ***Удалить верхний (нижний) колонтитул***.

Для тех, кто работает с LibreOffice

Для создания колонтитулов используются команды **Верхний колонтитул** и **Нижний колонтитул** меню **Вставка**. После выбора одной из этих команд, а также стиля колонтитула пользователь может отредактировать содержимое колонтитулов (рис. 3.49).



Рис. 3.49. Верхний колонтитул

Для вставки в колонтитул таких значений свойств документа, как номер страницы, текущая дата, количество страниц, данные об авторе и др., следует использовать команды кнопки со списком **Вставить поле** панели инструментов **Стандартная**. Для завершения редактирования документа следует выбрать место за пределами колонтитула.

Для использования различных колонтитулов на разных страницах документа нужно выполнить ***Формат*** ⇒ ***Страница***, в окне **Стиль страницы** выбрать вкладку **Верхний (нижний) колонтитул** и установить необходимые значения.

СТИЛИ В ТЕКСТОВОМ ДОКУМЕНТЕ

На предыдущих уроках вам приходилось неоднократно создавать новые текстовые документы и оформлять их должным образом, устанавливая нужные значения свойств для различных типов объектов – текста, таблиц, рисунков, формул, объектов **SmartArt** и т. п. Если документ большой и имеет значительное количество различных объектов, то на форматирование каждого отдельного объекта понадобится значительное время. Тем более, если часто случаются объекты, которые должны быть отформатированы одинаково, – следить за одинаковым оформлением однотипных элементов документа достаточно сложно.

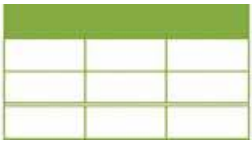
Для ускорения оформления документов и их объектов удобно использовать стили. **Стиль – это набор значений свойств определенного типа объектов, который имеет собственное имя.**

Текстовый процессор **Word 2007** по умолчанию имеет собственную библиотеку профессионально разработанных стилей, так называемые **экспресс-стили**, в которых значения свойств объектов подобраны гармонично, с учетом основных требований к дизайну. Такие стили разработаны для различных типов объектов текстового документа – текста, таблиц, графических объектов и т. п. (табл. 3.10). Каждый экспресс-стиль в библиотеке имеет имя. Вы уже применяли их в форматировании графических изображений, таблиц.

Использование стилей позволяет одним действием изменить сразу значения нескольких свойств объекта. Например, для того чтобы вручную отформатировать текст с такими значениями свойств, которые имеет стиль **Заголовок 4** (табл. 3.11), нужно выполнить 8 отдельных действий – вы-

Таблица 3.11

Примеры стилей разных объектов в Word 2007

Тип объекта	*Имя стиля*	*Образец*	*Некоторые значения свойств*
Текст	Заголовок 4	*Для быстрого оформления текста очень удобно использовать стили*	Шрифт – *Cambria* Размер – *12 пт* Цвет – *черный* Начертание – *полужирный курсив* Междустрочный интервал – *1,5* Отступ первой строки – *нет* Выравнивание – *по ширине*
Таблица	Светлый список акцент 6		Цвет линий – *зеленый* Цвет заливки – *зеленый* Тип линии – *сплошная* Толщина линии – *0,5 пт*
Фигура	Центральный градиент – Акцент 2		Цвет – *светло-коричневый* Прозрачность – *0 %* Тип линии – *сплошная* Толщина линии – *1 пт* Цвет контура – *темно-коричневый* Теневые эффекты – *внизу справа*

брать соответствующие элементы управления на вкладках или мини-панели для установки значений каждого свойства, при применении стиля **Заголовок 4** – только одно действие для выбора стиля из библиотеки.

Особенно обратите внимание на такие стили текста, как **Заголовок 1, Заголовок 2, ..., Заголовок 9,** которые позволяют автоматизировать создание структуры документа.

Коллекцию стилей пользователь может дополнять собственными стилями, создавая их на основе уже существующих или заново. При сохранении документа с ним автоматически сохраняются и примененные стили, то есть при дальнейших открытиях документа внешний вид документа не будет меняться.

ИСПОЛЬЗОВАНИЕ ЭКСПРЕСС-СТИЛЕЙ

Эскизы экспресс-стилей текстовых объектов документа (символов, абзацев) отображены на вкладке **Главная** в группе **Стили** (рис. 3.50).

Каждый эскиз из этого списка имеет оформление, соответствующее выбранному стилю, а рядом с именем стиля содержится значок, показывающий тип объектов, к которым этот стиль применяется: к абзацам – **¶**; к символам – **a**; к символам и абзацам – **¶a**. Стили заголовков используются для форматирования названий структурных частей документа (разделов, пунктов, подпунктов и т. п.).



Рис. 3.50. Эскизы экспресс-стилей для текстовых объектов

В списке экспресс-стилей всегда отображается несколько эскизов, которые использовались последними. Для просмотра всех эскизов нужно использовать кнопки или полосы прокрутки списка. Для открытия всего списка нужно выбрать кнопку **Дополнительные параметры**.

Для применения экспресс-стиля нужно выделить объект и выбрать в списке нужный эскиз.

Стили можно переименовывать или удалять, воспользовавшись командами контекстного меню эскиза стиля. Следует помнить, что указанные операции невозможны над стилями заголовков.

Для отказа от применения экспресс-стиля и установки значений свойств объекта по умолчанию нужно выбрать команду **Очистить формат**.

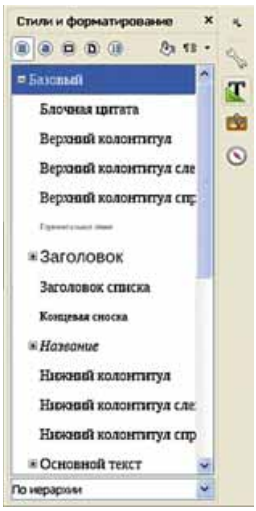


Рис. 3.51. Панель **Стили и форматирование**

Для тех, кто работает с LibreOffice

Для использования стилей нужно открыть панель **Стили и форматирование** (рис. 3.51) в правой части окна программы (***Формат*** ⇒ ***Стили и форматирование***) или выбрать кнопку **Стили и форматирование** на правой боковой панели окна программы.

Выбор соответствующих кнопок открывает список стилей: – стили абзацев, – стили символов, – стили страницы, – стили списков.

Для тех, кто хочет знать больше

Темы документа и их применение

Стилевое оформление может быть применено не только к отдельным объектам, но и ко всему текстовому документу как единому целому.

Тема документа – это стиль оформления текстового документа, который имеет имя и определяет для его объектов значения некоторых свойств (цвет, размер и шрифт символов, способ выравнивания абзацев, междустрочный интервал, эффекты рисунков и т. п.). В пятом классе вы применяли темы для оформления презентаций.

В **Word 2007** есть библиотека встроенных тем, список которых открывается выбором кнопки **Тема** на вкладке **Разметка страницы** в группе **Темы** (рис. 3.52).

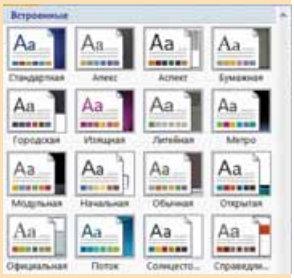


Рис. 3.52. Библиотека встроенных тем документа

Тему можно выбрать перед началом создания документа или в процессе его подготовки. Просмотр выбранной темы происходит динамически. Выбор темы приведет к изменению стилей оформления таких объектов документа, как текст, таблицы, графические изображения и т. п., с учетом того, что они были отформатированы с использованием стилей.

СОЗДАНИЕ СТРУКТУРЫ ДОКУМЕНТА И ЕЕ ИСПОЛЬЗОВАНИЕ

Создание структуры документа происходит с применением стилей **Заголовок 1, Заголовок 2, ..., Заголовок 9** – т. е. названия всех структурных частей документа следует отформатировать с использованием соответству-

ющих стилей. А именно, к названиям структурных частей первого уровня (разделов) применить стиль **Заголовок 1**, названия частей второго уровня (пунктов) отформатировать с применением стиля **Заголовок 2** и т. д.

Для этого следует:

1. Выделить название определенной структурной части документа (раздела, пункта, подпункта).
2. Выбрать на вкладке **Главная** в группе **Стили** такой стиль заголовка, который соответствует уровню структурной части в иерархической схеме документа (I уровень – **Заголовок 1**, II уровень – **Заголовок 2** и т. д.).
3. Повторить п. 1–2 этого алгоритма для всех структурных частей документа.

Для просмотра структуры документа, который отформатирован таким образом, используется режим просмотра **Структура**, который устанавливается выбором кнопки **Структура** в **Строке состояния** или на вкладке **Вид** в группе **Режимы просмотра документа**.

В этом режиме внешний вид документа отличается от его вида в режиме **Разметка страницы**, в котором вы привыкли работать (рис. 3.53). В режиме **Структура** текст структурных частей документа отображается в так называемом *свернутом виде* и на экран выводятся только заголовки структурных частей. Для лучшей визуализации иерархической структуры уровни документа имеют разный отступ от левого поля. Рядом с каждым уровнем документа есть соответствующий значок:

– если у уровня есть подуровни или вложенный текст;

– если у уровня отсутствует вложенный текст;

– если этот текст имеет самый низкий уровень (так называемый *основной текст*, к которому не применялись стили заголовков).

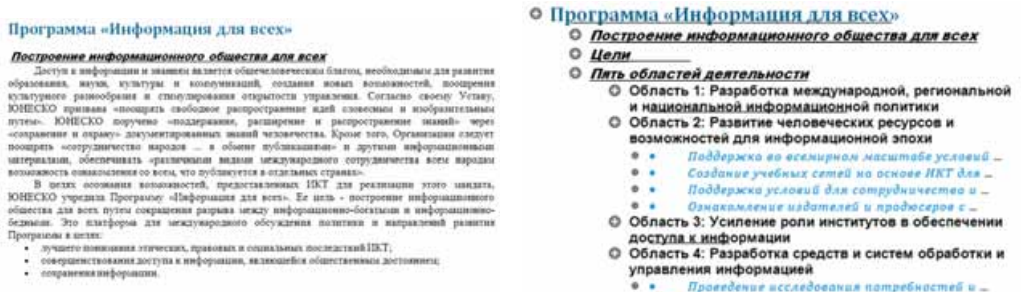


Рис. 3.53. Вид текста документа в режиме **Разметка страницы** и **Структура**

Включение режима **Структура** приводит к появлению на **Ленте** дополнительной вкладки **Структура** (рис. 3.54), элементы управления которой используются для отображения текста только нужных уровней документа, перемещения структурных частей в документе, изменения их уровней, удаления разделов текста или добавления новых и т. п.

Основное удобство просмотра документа в режиме **Структура** состоит в том, что существует возможность настраивать отображение любого уровня документа, скрывая при этом те уровни, которые занимают низшую иерархию. Для этого следует на вкладке **Структура** в группе **Работа со структурой**

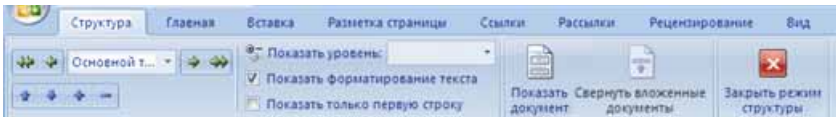


Рис. 3.54. Вкладка **Структура**

выбрать нужный уровень из списка **Показать уровень**. Если нужно развернуть содержание некоторой отдельной структурной части документа, следует дважды щелкнуть на значке ⊕ у соответствующего заголовка. Повторение этих действий скроет вложенный текст.

Если вложенный текст имеет слишком крупный шрифт или слишком длинный, мешает работе, можно отключить отображение форматирования (снять отметку флажка **Показать форматирование текста**), а также отображение всех строк текста, кроме первой (установить отметку флажка **Показать только строку**).

Самое важное в этом пункте

Структура документа – это иерархическая схема размещения составных частей документа. Чаще всего структуру создают для многостраничных документов, разделяя их на разделы, главы, параграфы, пункты и т. д.

Для того чтобы разделить текстовый документ на разделы в том месте, где будет начинаться новый раздел, нужно вставить символ **Разрыв раздела**. Это действие желательно выполнять в режиме просмотра документа **Разметка страницы** и при включенном режиме отображения скрытых (непечатаемых) символов. Далее нужно выполнить ***Разметка страницы*** ⇒ ***Параметры страницы*** ⇒ ***Разрывы*** и выбрать в списке **Разрывы разделов** нужный тип.

В месте разрыва разделов появляется скрытый (непечатаемый) символ **Разрыв раздела**. В документе, разделенном на разделы, можно выполнить разное форматирование отдельных частей документа.

В многостраничных документах часто используют ***колонтитулы*** – это служебные сообщения, которые размещаются на полях страницы и повторяются на других страницах документа. В **Word 2007** различают *верхний, нижний* и *боковые* колонтитулы. Они могут содержать номера страниц, название документа или текущего раздела, фамилию автора, графические изображения и т. п. Колонтитулы первой страницы, четных и нечетных страниц могут отличаться. Также могут различаться колонтитулы разных разделов документа.

В документе колонтитулы отображаются только в режимах **Разметка страницы** и **Режим чтения**. Обработка основного текста документа при работе с колонтитулами невозможна. Для создания колонтитулов используются элементы управления вкладки **Вставка** группы **Колонтитулы**.

Для ускорения оформления документов и их объектов удобно использовать стили. ***Стиль*** – это набор значений свойств определенного типа объектов, который имеет собственное имя. Текстовый процессор **Word 2007** имеет собственную библиотеку профессионально разработанных стилей, так называемые ***экспресс-стили***, в которых значения свойств объектов по-

добраны гармонично, с учетом основных требований к дизайну. Эскизы экспресс-стилей текстовых объектов документа отображены на вкладке **Главная** в группе **Стили.**

Создание структуры документа происходит с применением стилей **Заголовок 1**, **Заголовок 2**, ..., **Заголовок 9** – названия всех структурных частей документа следует отформатировать с использованием соответствующего стиля. Для просмотра структуры документа, отформатированного таким образом, используется режим просмотра **Структура**, который устанавливается выбором кнопки **Структура**  в **Строке состояния** или на вкладке **Вид** в группе **Режимы просмотра документа.**

Дайте ответы на вопросы

1°. Что такое структура документа? Приведите примеры структурированных документов.
2•. Что такое разделы? Для чего они применяются?
3•. Какие виды разрывов разделов можно использовать в текстовом документе?
4•. Что такое колонтитулы? Какие виды колонтитулов могут быть на странице?
5•. Каково назначение колонтитулов? Какие объекты можно вставить в колонтитулы?
6*. Как установить различные колонтитулы для разных страниц документа?
7°. Что такое стили? Для чего их используют?
8•. К каким объектам в **Word 2007** можно применять стилевое оформление? Какие свойства объектов документа они определяют?
9•. Каково назначение стилей заголовков? Сколько таких стилей в **Word 2007**?
10•. Для чего предназначен режим отображения документа **Структура**? Какие операции с документами в нем можно выполнять?

Выполните задания

1•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.5\образец.docx**). Каждый пункт текстового документа оформите как отдельный раздел, который начинается с новой страницы. Установите для разделов документа следующий формат:
а) нечетные разделы: ориентация страницы – *книжная*, поля – *по 2 см*, нумерация страниц – *сверху страницы по центру*;
б) четные разделы: ориентация страницы – *альбомная*, поля – *по 1 см*, нумерация страниц – *внизу страницы слева*.
Сохраните документ в вашей папке в файле **задание 3.5.1.docx**.

2•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.5\образец.docx**). Создайте в документе колонтитулы: в верхнем запишите текущую дату с выравниванием *по центру*, в нижнем – *свою фамилию* с выравниванием *по левому краю*. Сохраните документ в вашей папке в файле с именем **задание 3.5.2.docx**.

3*. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.5\образец.docx**). Отформатируйте документ следующим образом: размер страницы – *A4*; ориентация – *книжная*; поля: верхнее – *1,5 см*, нижнее – *1 см*, левое – *2 см*, правое – *3 см*; нумерация страниц – *внизу страницы справа*, начиная с номера *1*; колонтитулы: верхний на четных страницах – *название текстового документа*, нижний на четных страницах – *текущая дата и время*. Сохраните документ в папке **Мои документы** в файле с именем **задание 3.5.3.docx**.

4•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.5\образец.docx**). Отформатируйте объекты текстового документа с использованием следующих стилей:

- Заголовки – стиль *Название*.
- Текстовые фрагменты – стиль *Слабое выделение*.
- Списки – *маркированный* з маркером ◆.

Сохраните документ в вашей папке в файле с именем **задание 3.5.4.docx.**

5•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.5\образец1.docx**), в котором к объектам текстового документа было применено стилевое оформление. Выясните значения свойств объектов, которые устанавливают использованные стили. Запишите результаты в таблицу:

№ объекта	*Тип объекта*	*Стиль объекта*	*Значения свойств объекта*

6•. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.5\образец1.docx**), в котором к объектам текстового документа было применено стилевое оформление. Очистите форматы объектов документа. Запишите в таблицу значения свойств объектов по умолчанию:

№ объекта	*Тип объекта*	*Стиль объекта*	*Значения свойств объекта*

7*. Выясните с помощью **Справки**, как создать собственный стиль на основе отформатированного фрагмента документа. Составьте соответствующий алгоритм. Выполните это на практике.

8•. Откройте указанный учителем документ (например, **Глава 3\Пункт 3.5\образец.docx**). Отформатируйте документ, используя следующие стили:

- Заголовки I уровня – стиль *Заголовок 1*;
- Заголовки II уровня – стиль *Заголовок 2*;
- Заголовки III уровня – стиль *Заголовок 3*;
- Другие фрагменты текста – стиль *Обычный*.

Просмотрите структуру отформатированного документа. Сохраните документ в вашей папке в файле с именем **задание 3.5.8.docx.**

3.6. СРЕДСТВА АВТОМАТИЗАЦИИ СОЗДАНИЯ ТЕКСТОВОГО ДОКУМЕНТА

1. Что такое стили? Для чего их используют?
2. Что такое структура документа? Как ее отобразить?
3. Что такое гиперссылки? Для чего и где они используются?

ССЫЛКА В ТЕКСТОВОМ ДОКУМЕНТЕ

В предыдущем пункте вы ознакомились с некоторыми особенностями обработки многостраничных документов – просмотром структуры документа, созданием разделов, использованием стилевого оформления. Кроме этого, в текстовом процессоре **Word 2007** есть и другие средства для упрощения процесса создания больших документов и перемещения по ним.

Обычно в многостраничных документах встречаются различные виды *ссылок – специальных записей в документе, которые указывают на какие-либо объекты этого или других документов*. Ссылки предназначены для смыслового связывания одних объектов документа с другими. Например, в тексте могут находиться ссылки на определенный рисунок или таблицу, ссылки на библиографические источники, сайты или другое. При выборе этих ссылок происходит переход на указанный объект.

Текстовый процессор **Word 2007** позволяет автоматизировать процесс создания ссылок в документе. Инструменты, которые используются для этого, находятся на вкладке **Ссылки**, например:

- группа **Оглавление** предназначена для создания ***оглавления документа – перечня названий структурных частей документа с указанием номеров соответствующих страниц***;
- группа **Предметный указатель** предназначена для создания ***предметного (алфавитного) указателя – алфавитного списка терминов, использованных в документе, с указанием номеров страниц, где они встречаются.***

АВТОМАТИЗИРОВАННОЕ СОЗДАНИЕ ОГЛАВЛЕНИЯ ДОКУМЕНТА

Если для каждого из структурных частей многостраничного документа использовано стилевое форматирование с применением стилей заголовков, то оглавление документа можно создать автоматически. Выполняется эта операция в режиме просмотра документа **Разметка страницы**  .

Для этого нужно выполнить следующий алгоритм:

1. Установить курсор в том месте документа, где нужно разместить оглавление.
2. Выполнить ***Ссылки*** ⇒ ***Оглавление*** ⇒ ***Оглавление***  .
3. Выбрать в списке встроенных образцов нужный вариант оформления оглавления (рис. 3.55).

Оглавление будет представлено в виде таблицы, в которую внесены названия структурных частей текстового документа и соответствующие номера страниц, где этот фрагмент текста начинается (рис. 3.56). Обычно оглавление размещают в начале документа или в его конце.

Созданное оглавление можно использовать для быстрого перемещения по документу, поиска нужных структурных частей – разделов, пунктов и т. п. Для этого следует в оглавлении документа выбрать нужный заголовок при нажатой клавише **Ctrl**. Вид текстового курсора при этом изменится на такой: .

Если в ходе работы над документом его текст и структура изменялись, то оглавление документа нужно обновить. Для этого следует выполнить ***Ссылки*** ⇒ ***Оглавление*** ⇒ ***Обновить таблицу*** .

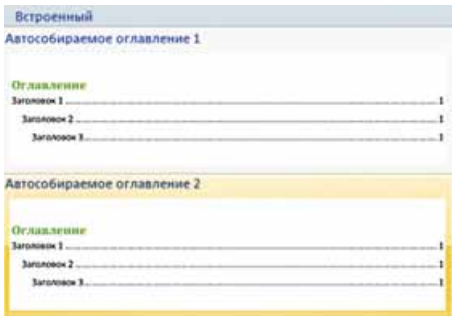


Рис. 3.55. Список образцов оформления оглавлений

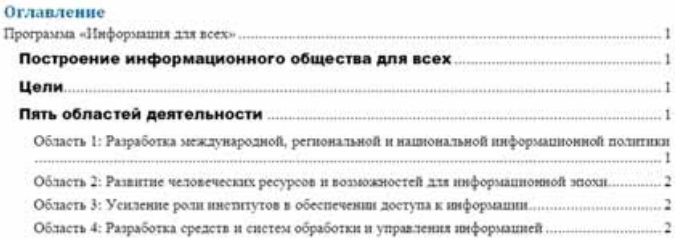
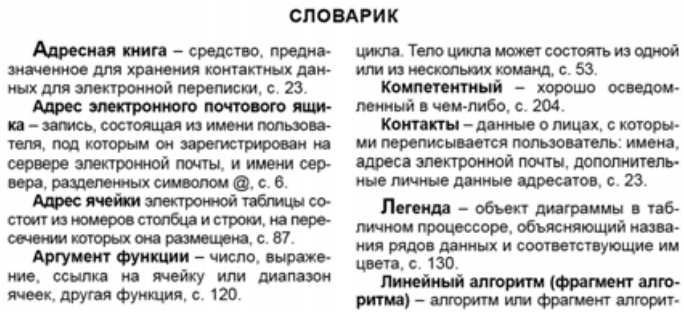


Рис. 3.56. Пример содержания документа

АВТОМАТИЗИРОВАННОЕ СОЗДАНИЕ ПРЕДМЕТНОГО УКАЗАТЕЛЯ

Предметные указатели чаще всего используются в учебниках, научных изданиях, словарях. Размещаются они обычно в конце документа и выглядят примерно так (рис. 3.57):


СЛОВАРИК

Адресная книга – средство, предназначенное для хранения контактных данных для электронной переписки, с. 23.

Адрес электронного почтового ящика – запись, состоящая из имени пользователя, под которым он зарегистрирован на сервере электронной почты, и имени сервера, разделенных символом @, с. 6.

Адрес ячейки электронной таблицы состоит из номеров столбца и строки, на пересечении которых она размещена, с. 87.

Аргумент функции – число, выражение, ссылка на ячейку или диапазон ячеек, другая функция, с. 120.

цикла. Тело цикла может состоять из одной или из нескольких команд, с. 53.

Компетентный – хорошо осведомленный в чем-либо, с. 204.

Контакты – данные о лицах, с которыми переписывается пользователь: имена, адреса электронной почты, дополнительные личные данные адресатов, с. 23.

Легенда – объект диаграммы в табличном процессоре, объясняющий названия рядов данных и соответствующие им цвета, с. 130.

Линейный алгоритм (фрагмент алгоритма) – алгоритм или фрагмент алгорит-

Рис. 3.57. Пример предметного указателя в учебнике

Создание предметного указателя выполняется в два этапа: сначала нужно выделить все термины, которые следует поместить в указатель, затем вставить указатель в документ. Рассмотрим этапы более подробно.

I э т а п. Выделение элементов указателя:

1. Выделить термин или определение, которые нужно поместить в указатель, на странице, где они встречаются впервые.

2. Выполнить ***Ссылки*** ⇒ ***Предметный указатель*** ⇒ ***Пометить элемент***.
3. Установить в окне **Определение элемента указателя** (рис. 3.58) нужные значения:
   - *основной* – при необходимости можно изменить основное название термина;
   - *дополнительный* – при необходимости ввести дополнительные названия термина;
   - *перекрестная ссылка* – выбрать название другого термина, с которым связан данный;
   - *текущая страница* – снести в указатель для данного термина только номер текущей страницы;
   - *диапазон страниц* – снести в указатель для данного термина диапазон страниц, где он встречается (дополнительно нужно установить закладки для начальной и конечной страницы диапазона);
   - *формат номера страницы* – установить начертание номеров страниц в указателе.

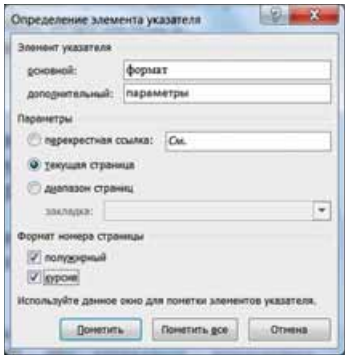


Рис. 3.58. Окно **Определение элемента указателя**

4. Выбрать кнопку **Пометить** (или **Пометить все**, если нужно, чтобы в указателе указывались все номера страниц, где этот термин встречается).
5. Повторить действия 1–4 для всех элементов указателя.

В результате таких действий все термины для предметного указателя будут определены и в тексте документа рядом с этими словами будет вставлено поле такого вида: **формат**{ XE "формат" }, однако отображаться оно будет только в режиме отображения скрытых (непечатаемых) символов.

II э т а п. Вставка предметного указателя в текстовый документ:

1. Установить курсор в то место документа, где нужно разместить указатель (обычно в конце документа).
2. Выполнить ***Ссылки*** ⇒ ***Предметный указатель*** ⇒ ***Указатель***.
3. Установить в окне **Указатель** (рис. 3.59) параметры отображения указателя:
   - *Номер страницы по правому краю* – при желании установить такой способ выравнивания номеров страниц в указателе;
   - *Форматы* – выбрать из предложенного списка желаемый вид шаблона указателя;
   - *Тип* – выбрать один из двух способов размещения номеров страниц у элемента указателя;
   - *Колонки* – указать количество колонок, в которых будет размещен указатель;
   - *Язык* – алфавит, по которому будут упорядочены элементы указателя.



Рис. 3.59. Окно **Указатель**

4. Выбрать кнопку **ОК**.

Таким образом предметный указатель будет вставлен в выбранное место в документе. Перед указателем и после него в документе будут вставлены скрытые символы **Разрыв раздела**.

Если некоторые данные, входящие в указатель, изменились (изменилось определение или размещение термина в документе), то содержание указателя нужно обновить. Для этого следует выделить указатель и выполнить ***Ссылки*** ⇒ ***Указатель*** ⇒ ***Обновить указатель*** .

Для удаления указателя нужно выделить его текст и две строки – до и после него со скрытыми символами разрывов разделов, затем нажать клавишу **Delete**.

Для тех, кто работает с LibreOffice

Для работы с указателями и оглавлением документа используются команды **Оглавление и указатели** меню **Вставка** (рис. 3.60).



Рис. 3.60. Список команд **Оглавление и указатели** меню **Вставка**

ШАБЛОНЫ ТЕКСТОВЫХ ДОКУМЕНТОВ

Вы уже научились создавать текстовые документы и форматировать их с использованием различных средств текстового процессора. Вместе с тем текстовый процессор **Word** для создания документов определенных типов предоставляет еще одну возможность – создание документа на основе шаблона. Вы уже создавали в пятом классе презентации с помощью шаблонов.

Шаблон **– это отформатированный определенным образом документ-заготовка, который сохранен в отдельном файле и используется как образец для создания новых документов определенного типа.**

Например, шаблон для создания календаря, благодарности, письма, отчета, поздравительной открытки, визитки и т. п.

Открыв шаблон, пользователь видит на экране текстовый документ, в котором будут содержаться пустые поля для ввода текста, надписей, вставки рисунков, колонтитулов, логотипов и т. п. К этому документу уже могут быть применены определенные темы и стили оформления различных объектов (рис. 3.61).

Пользователю остается только заполнить соответствующие места документа нужным текстом и сохранить документ в текстовом файле – форматирование и структурирование документа уже определены самим шаблоном.

Использование шаблонов значительно облегчает работу пользователя, поскольку они являются основой для быстрого создания аналогичных по форме, но различных по содержанию документов без затрат времени на их форматирование.

Основное различие между документами и шаблонами состоит в их использовании: шаблон – это заготовка документа с готовыми элементами текста и их оформления, которая предназначена для дальнейшего заполнения данными, а документ – это уже готовый текст, возможно, на основе некоторого шаблона.



Рис. 3.61. Шаблон меню праздника и созданный на его основе документ

Шаблоны хранятся в файлах с расширениями **dotx** и **dotm**. *Помните*, в шаблонах могут храниться макровирусы, поэтому в работе используйте шаблоны, которые вы получили из надежного источника.

В текстовом процессоре **Word 2007** все шаблоны делятся на три группы:

- *установленные* – шаблоны документов определенных типов (писем, факсов, отчетов и т. п.), установленные на компьютере в составе пакета **Microsoft Office 2007**;
- *Microsoft Office Online* – шаблоны документов различных типов, которые расположены на сайте **Microsoft Office Online**;
- *шаблоны пользователя* – шаблоны, созданные пользователем.

Основным из стандартных шаблонов **Word 2007** является шаблон документов ***Обычный*** (хранится в файле **Normal.dotm**), который автоматически открывается при запуске программы **Word 2007** и устанавливает по умолчанию такой формат документа (табл. 3.12):

Таблица 3.12

Значения свойств объектов документа в шаблоне Normal.dotm

Страница	*Абзацы*	*Символы*
• Ориентация страницы – *книжная* • Размер страницы – *A4* • Верхнее поле – *1,5 см* • Нижнее поле – *1 см* • Левое поле – *2 см* • Правое поле – *1,5 см*	• Междустрочный интервал – *одинарный* • Интервал после абзаца – *10 пт* • Выравнивание – *по левому краю* • Отступы – *отсутствуют*	• Шрифт основного текста – *Calibri* • Размер – *11 пт* • Цвет – *черный* • Интервал – *обычный*

Также этот шаблон определяет стилевое оформление заголовков, списков, таблиц и т. п. Ввод текста, форматирование отдельных объектов, структури-

рование текста пользователь осуществляет самостоятельно. На основе шаблона **Normal.dotm** можно создавать документы различных типов.

Еще одним из стандартных шаблонов **Word 2007** является, например, шаблон **Обычное резюме**. Этот шаблон содержит в документе-заготовке несколько текстовых полей, в которые пользователь вносит данные о себе: имя и фамилию, контактную информацию, сведения об образовании, место работы и т. п. (фрагмент шаблона приведен на рис. 3.62). Вставленную фотографию заменяет на собственную. Дату можно выбрать в календаре, который открывается при выборе указанного поля. Каждая структурная часть документа уже отформатирована и размещена в тексте соответствующим образом. Используя этот шаблон, можно быстро подготовить собственное резюме.

Резюме (франц. *résumé*) – вид документа, в котором подаются короткие сведения об учебе, трудовой деятельности и профессиональных успехах и достижениях лица, которое его составляет.

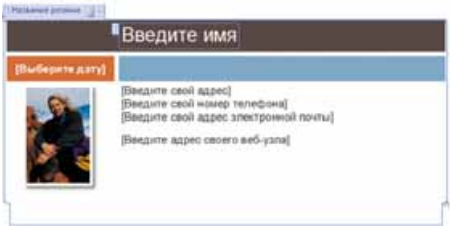


Рис. 3.62. Шаблон **Обычное резюме**

СОЗДАНИЕ ТЕКСТОВЫХ ДОКУМЕНТОВ НА ОСНОВЕ ШАБЛОНОВ

Для того чтобы в текстовом процессоре **Word 2007** создать документ на основе установленных шаблонов, нужно:

1. Выбрать кнопку **Office**.
2. Выбрать команду **Создать**, которая открывает диалоговое окно **Создание документа**.
3. Выбрать в списке слева в разделе **Шаблоны** нужную группу шаблонов – **Установленные шаблоны** (рис. 3.63).
4. Выбрать в списке шаблонов требуемый (например, **Обычное резюме**).
5. Просмотреть структуру и внешний вид шаблона в поле образцов (справа в окне).

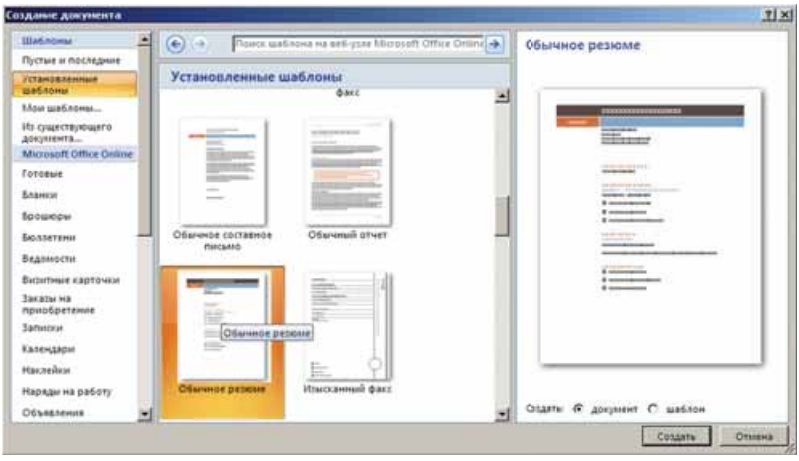


Рис. 3.63. Создание документа на основе инсталлированного шаблона

6. Установить в нижней части поля образцов переключатель **Документ**.
7. Выбрать кнопку **Создать**.
8. Заполнить предложенные поля нужными данными.
9. Сохранить документ.

Если пользователь желает применить один из шаблонов, которые размещены на сайте **Microsoft Office Online**, то следует выполнить следующие действия:

1. Выбрать кнопку **Office**.
2. Выбрать команду **Создать**, которая открывает диалоговое окно **Создание документа**.
3. Выбрать в списке слева в разделе **Microsoft Office Online** нужный тип шаблона (рис. 3.64). Например, **Открытки**.

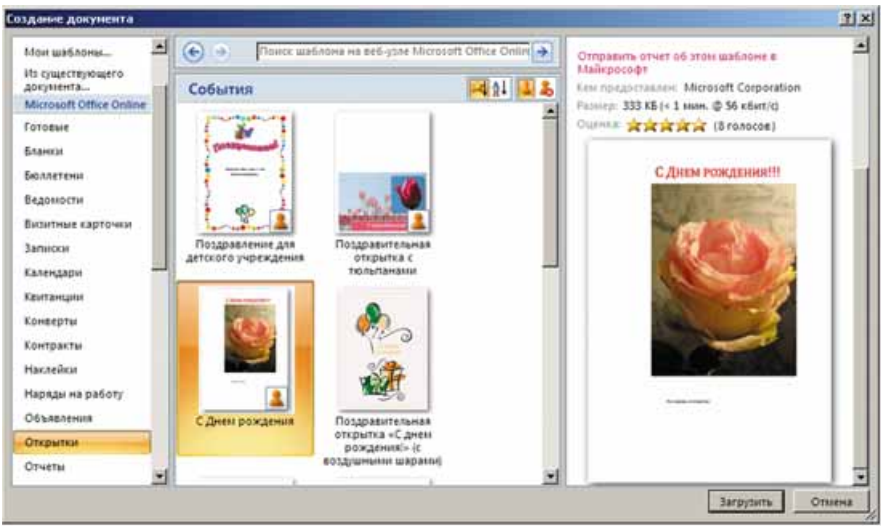


Рис. 3.64. Создание документа на основе шаблона из Интернета

4. Дождаться соединения с сервером сайта и отображения списка шаблонов.
5. Выбрать нужный шаблон.
6. Просмотреть структуру и внешний вид шаблона в поле образцов.
7. Выбрать кнопку **Загрузить**.
8. Дождаться загрузки шаблона.
9. Заполнить, при необходимости, нужными данными предложенные поля.
10. Сохранить документ.

Для тех, кто работает с Microsoft Word 2010

В **Word 2010** для создания документа на основе шаблона необходимо выполнить ***Файл*** ⇒ ***Создать*** ⇒ ***Образцы шаблонов***, выбрать нужный шаблон, установить переключатель **Документ** и выбрать кнопку **Создать**.

Для выбора шаблона с сайта **Office.com** нужно выбрать требуемую категорию из раздела **Шаблоны Office.com** и загрузить выбранный шаблон.

Для тех, кто работает с LibreOffice

Создание документов на основе шаблонов осуществляется с использованием элементов управления окна **Управление шаблонами** (***Файл*** ⇒ ***Шаблоны*** ⇒ ***Управление***) (рис. 3.65).

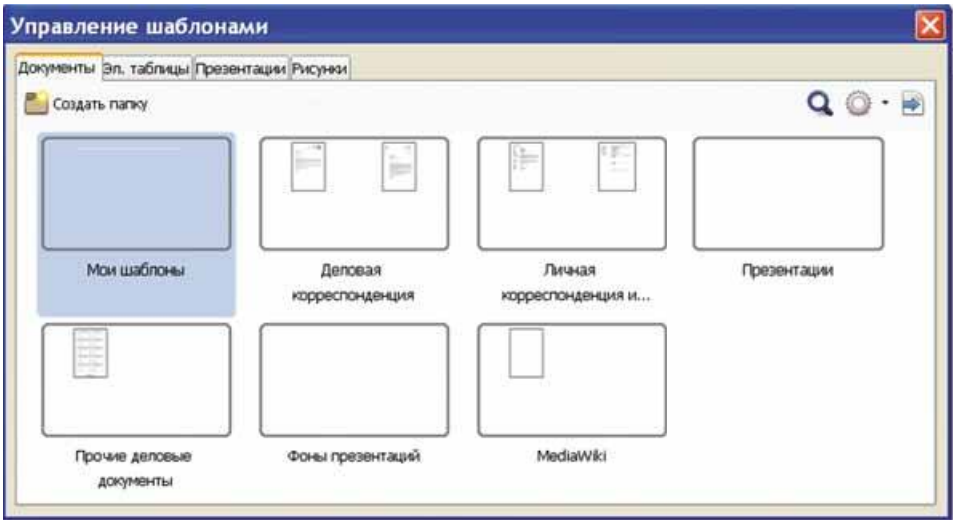


Рис. 3.65. Окно **Управление шаблонами**

РАБОТА С НЕСКОЛЬКИМИ ДОКУМЕНТАМИ

Текстовый процессор **Word 2007** предоставляет пользователю возможность открыть для работы одновременно несколько документов. Это позволяет пользователю просматривать документы, сравнивать их содержание, вставлять фрагменты одного документа в другой, объединять их и т. п.

Для управления открытыми окнами нескольких документов в программе существуют специальные средства, которые расположены на вкладке **Вид** в группе **Окно** (рис. 3.66). Так, переключение между окнами можно осуществить, выполнив следующий алгоритм:

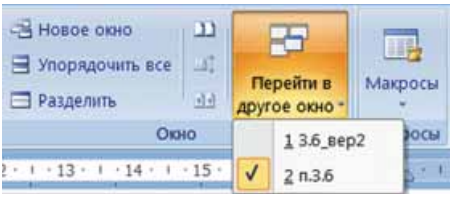


Рис. 3.66. Свернутая группа **Окно**

1. Выбрать кнопку **Перейти в другое окно** в группе **Окно** на вкладке **Вид**.
2. В списке открытых окон выбрать нужное (текущий документ отмечен √).

Когда работа происходит с двумя различными версиями одного и того же документа, удобно отобразить их на экране рядом. Для выполнения этой операции нужно выполнить ***Вид*** ⇒ ***Окно*** ⇒ ***Рядом*** и выбрать в списке открытых документов нужный. После этого экран будет разделен вертикально пополам и в каждой из этих частей будет отображено окно

указанных документов. Прокрутка документов в двух окнах будет происходить синхронно.

Чтобы отменить синхронную прокрутку, нужно выбрать соответствующую кнопку в группе **Окно** (рис. 3.67).

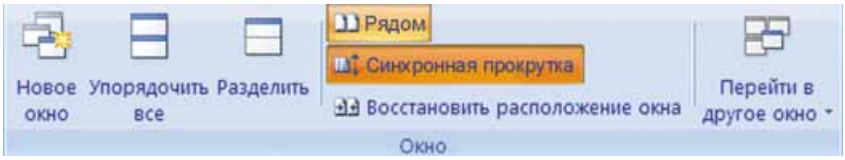


Рис. 3.67. Развернутая группа **Окно**

Для отмены режима отображения документов рядом следует повторно выбрать кнопку **Рядом** в группе **Окно.**

Для тех, кто хочет знать больше

Во время обработки документа иногда возникает потребность видеть одновременно два разных его фрагмента и просматривать их независимо. Для этого удобно разделить окно документа на две части, выполнив последовательность действий ***Вид*** ⇒ ***Окно*** ⇒ ***Разделить***. На экране появится *полоса разделения окна*, которую следует перетащить в нужное место. Эту же операцию можно выполнить и перетаскиванием полосы разделения с помощью соответствующей метки, которая находится над вертикальной полосой прокрутки (рис. 3.68).

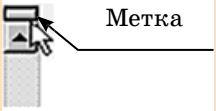


Рис. 3.68. Метка полосы разделения окна

Для возврата к работе в одном окне нужно выбрать кнопку **Снять разделение** в группе **Окно** вкладки **Вид** или дважды щелкнуть на полосе разделения окна.

ФОРМАТЫ ФАЙЛОВ ТЕКСТОВЫХ ДОКУМЕНТОВ

Стандартным форматом файлов текстовых документов, подготовленных в текстовом процессоре **Word 2007** и последующих версий, является формат **DOCX**. Кроме того, **Word 2007** обеспечивает работу с документами, сохраненными в других форматах, например **RTF**, **TXT**, **DOC**, **PDF** и т. п. Характеристика этих текстовых форматов приведена в таблице 3.13.

Таблица 3.13

Характеристика наиболее распространенных форматов файлов текстовых документов в Word 2007

Формат	*Стандартный значок*	*Характеристика*
TXT		В файле хранится только текст с разбивкой на абзацы, без форматирования. Может обрабатываться различными системами обработки текстов.
DOC		В файле хранятся текст, рисунки, вставленные объекты, значения их свойств. Является основным форматом для текстовых документов, подготовленных в версиях **Word** до 2003 включительно.

Продолжение таблицы 3.13

Формат	*Стандартный значок*	*Характеристика*
DOCX		В файле хранятся текст, рисунки, вставленные объекты, значения их свойств. Обеспечивает значительное сжатие данных по сравнению с другими форматами (до 70 %). Впервые использован в **Word 2007**.
RTF		В файле хранятся текст, рисунки, вставленные объекты, значения их свойств. Используется различными системами обработки текстов для разных ОС.
PDF		В файле полностью хранится форматирование документа, редактировать файл невозможно. Для просмотра файла используется **Adobe Reader**.

Для того чтобы сохранить текстовый документ не в стандартном формате, следует в окне **Сохранение документа** открыть список **Тип файла** и выбрать нужный формат.

Для тех, кто работает с LibreOffice

Стандартным форматом файлов текстовых документов, подготовленных в текстовом процессоре **LibreOffice Writer**, есть формат **ODF** (англ. *Open Document Format for Office Application* – открытый формат документов для офисных программ) с расширением имени файла **ODT**. Кроме того, **LibreOffice Writer** обеспечивает работу с документами, сохраненными в других форматах, например **RTF**, **TXT**, **DOC**, **XML** и т. п.

Самое важное в этом пункте

В больших многостраничных документах в **Word 2007** можно создать различные виды ***ссылок*** – ***специальные записи, которые указывают на некоторые объекты того или иного документа.*** Ссылки предназначены для логического связывания одних объектов документа с другими. При выборе ссылки в документе происходит переход на указанный объект.

Инструменты, которые предназначены для создания ссылок, находятся на вкладке **Ссылки**. С их использованием можно создать, например, оглавление документа (группа **Оглавление**) и предметный указатель терминов (группа **Указатель**).

Для создания документов определенных типов в текстовом процессоре **Word** можно использовать ***шаблоны*** – ***отформатированные определенным образом документы-заготовки, которые сохранены в отдельных файлах и используются в качестве образца для создания новых документов.*** Шаблоны хранятся в файлах с расширениями **dotx** и **dotm**.

Для того чтобы в текстовом процессоре **Word 2007** создать документ на основе шаблона, нужно выполнить ***Office*** ⇒ ***Создать*** ⇒ ***Шаблоны*** ⇒ выбрать нужный шаблон ⇒ ***Создать***.

Текстовый процессор **Word 2007** предоставляет пользователю возможность одновременно работать с несколькими открытыми документами. Это позволяет пользователю просматривать документы, сравнивать их содер-

жимое, вставлять фрагменты одного документа в другой, объединять их и т. п. Для управления открытыми окнами в программе существуют специальные средства, которые расположены на вкладке **Вид** в группе **Окно**.

Стандартным форматом файлов текстовых документов, подготовленных в текстовом процессоре **Word 2007**, является формат **DOCX**. Кроме того, **Word 2007** обеспечивает работу с документами, сохраненными в других форматах, например **RTF**, **TXT**, **DOC**, **PDF** и т. п.

Дайте ответы на вопросы

1•. Что такое ссылки в документе? Для чего они используются?
2°. Что такое оглавление документа? Для чего оно предназначено?
3•. Какие средства **Word 2007** используют для создания оглавления документа?
4•. Что такое предметный указатель? Для чего он используется?
5•. Как создать предметный указатель в текстовом документе?
6°. Что такое шаблон? Для чего его используют?
7•. Какие свойства документа может устанавливать шаблон? Какие типы шаблонов используют в **Word 2007**?
8•. Чем шаблон отличается от документа?
9•. Какое расширение имеет файл, в котором хранится шаблон? Приведите примеры шаблонов.
10•. Какие существуют способы одновременного отображения на экране окон открытых документов? Как их установить?
11°. В каких форматах может быть сохранен файл текстового документа в **Word 2007**? Какой формат является стандартным?
12•. Как выбрать формат для сохранения файла текстового документа?

Выполните задания

1•. Откройте указанный учителем документ (например, **Глава 3\Пункт 3.6\образец.docx**). Отформатируйте названия структурных частей документа с использованием стилей заголовков соответствующих уровней. Создайте в начале документа его оглавление. Сохраните документ в вашей папке в файле с именем **задание 3.6.1.docx**.
2•. Откройте указанный учителем документ (например, **Глава 3\Пункт 3.6\образец.docx**). Создайте в конце документа предметный указатель для терминов, которые выделены в тексте зеленым цветом, и их определений. Формат указателя установите по собственному желанию. Сохраните документ в вашей папке в файле с именем **задание 3.6.2.docx**.
3•. Откройте текстовый процессор **Word 2007**. Просмотрите установленные шаблоны документов. Выясните их количество, тематику документов, которые можно создать на их основе. Результаты запишите в тетрадь.
4*. Найдите на вашем компьютере файлы шаблонов документов. Запишите в тетрадь: в какой папке они сохранены? Сколько таких файлов-шаблонов? Какие их имена?
5•. Создайте в текстовом документе письмо на основе шаблона *Изысканное письмо* в оргкомитет конкурса «Україна має талант» о же-

лании принять в нем участие. Сохраните документ в вашей папке в файле с именем **задание 3.6.5.docx**.

6•. Создайте в текстовом документе собственную визитку, воспользовавшись соответствующим шаблоном с сайта **Microsoft Office Online**. Сохраните документ в вашей папке в файле с именем **задание 3.6.6.docx**.

7*. С помощью **Справки** выясните, как создать шаблон на основе существующего документа. Создайте таким образом собственный шаблон оформления отчета о прохождении летней практики. Сохраните шаблон в коллекции.

8•. Откройте в текстовом процессоре **Word 2007** три текстовых файла (например, из папки **Глава 3\Пункт 3.6\образец1.docx, образец2.docx, образец3.docx**). Обработайте переключение между окнами с помощью кнопок этих окон на **Панели задач** и кнопки **Переход в другое окно**. Вставьте в начале первого документа первый абзац второго документа, а в конце первого документа – последний абзац третьего документа. Сохраните первый документ в файле с именем **задание 3.6.8.docx** в вашей папке.

9•. Откройте в текстовом процессоре **Word 2007** два указанных учителем файла (например, **Глава 3\Пункт 3.6\образец1.docx, образец2.docx**). Отобразите окна этих документов рядом. Обработайте перемещение по документам при включенном и выключенном режиме **Синхронная прокрутка**. Найдите отличия в первом документе по сравнению со вторым и выделите их цветным фоном. Снимите режим просмотра документов рядом. Сохраните первый документ в файле с именем **задание 3.6.9.docx** в вашей папке.

10•. Откройте любой файл формата **DOCX** (например, **Глава 3\Пункт 3.6\образец.docx**). Сохраните его в вашей папке в формате **DOC** и **TXT**. Просмотрите созданные файлы. Сравните размеры трех файлов. Объясните полученный результат.

ПРАКТИЧЕСКАЯ РАБОТА № 5

«Структура документа. Автоматизированное создание оглавления и предметного указателя»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте указанный учителем файл (например, **Глава 3\Пункт 3.6\Практическая5\образец.docx**).
2. В режиме просмотра **Структура** упорядочите структурные части документа в правильном порядке. Измените при необходимости уровень структурных элементов.
3. Отформатируйте названия структурных частей текстового документа с использованием стилей заголовков соответствующего уровня.
4. Создайте в начале документа его оглавление.
5. Создайте в конце документа предметный указатель для всех слов, которые обозначены красным цветом.
6. Сохраните документ в вашей папке в файле с именем **практическая работа 5.docx**.

Глава 4. Обработка объектов мультимедиа

В этой главе вы узнаете:

- о форматах аудио- и видеофайлов
- о программном обеспечении для обработки объектов мультимедиа
- о средствах преобразования аудио- и видеоформатов
- о средствах записи/захвата аудио и видео
- о создании аудио- и видеофрагментов с использованием:
  - аудио- и видеорядов с согласованием по времени
  - видеоэффектов
  - переходов между отдельными частями видеофрагментов
- о сервисах размещения аудио- и видеофайлов в Интернете



4.1. ОБРАБОТКА ОБЪЕКТОВ МУЛЬТИМЕДИА

1. Какие сообщения называют мультимедийными?
2. Какие программы для обработки мультимедийных объектов вы знаете? Для чего предназначены проигрыватели мультимедийных данных?
3. Какие устройства используются для создания и обработки мультимедийных сообщений?

ФОРМАТЫ АУДИО- И ВИДЕОФАЙЛОВ

Ранее вы уже ознакомились с понятием типа (формата) файла. Тип (формат) файла определяет структуру данных в файле. Вам известны различные форматы текстовых и графических файлов, файлов-презентаций и электронных таблиц. Аудио- и видеоданные также хранятся в файлах определенных форматов, эти файлы, как вам известно, называют мультимедийными. Примеры форматов аудио- и видеофайлов с описанием их особенностей представлены в таблице 4.1.

Таблица 4.1

Типы (форматы) мультимедийных файлов

Тип (формат)	*Описание*	*Расширение имени*
Файлы, в которых не используется сжатие или используется сжатие без потери данных		
WAV или **WAVE** (англ. ***waveform audio format*** – аудиоформат типа *волна*)	Формат звуковых файлов, используется как основной для несжатых аудиоданных в операционных системах семейства **Windows**	**wav**
MIDI (англ. ***Musical Instrument Digital Interface*** – цифровой интерфейс музыкальных инструментов)	Формат файлов, содержащих команды для воспроизведения звука специальным устройством или программой – синтезатором	**mid, midi**
FLAC (англ. ***Free Lossless Audio Codec*** – свободный аудиокодек без потерь)	Формат для сжатия аудиоданных без потерь, распространяется по лицензии свободного программного обеспечения	**flac**
Файлы, в которых используется сжатие с частичной потерей данных		
MP3, точнее **MPEG-1/2/2.5 Layer 3** (англ. ***Motion Picture Experts Group*** – экспертная группа по вопросам движущегося изображения, ***Layer 3*** – третий уровень)	Один из самых распространенных форматов аудиофайлов. Используется для передачи звуковых данных в глобальных сетях, в файлообменных системах	**mp3**
MP4 или **MPEG-4 Part 14**	Файлы этого формата обычно используются для цифрового телевидения, хранения видеофильмов и т. п.	**mp4**

Продолжение таблицы 4.1

Тип (формат)	*Описание*	*Расширение имени*
WMA/WMV (англ. *Windows Media Audio/Video*)	Формат файлов для хранения и воспроизведения аудио-, видеоданных, разработанный корпорацией **Microsoft** и стандартно используется в ОС **Windows**	**wma, wmv**
QuickTime (англ. ***Quick Time*** – быстрое время)	Формат файлов, основанный на технологии хранения и воспроизведения видео- и звуковых данных, разработанной корпорацией **Apple**	**mov**
Flash видео	Формат файлов, используемых обычно для распространения видео в Интернете. Характеризуется высокой степенью сжатия данных	**flv**

В связи с тем что большинство аудио- и видеоданных сжимаются с использованием различных алгоритмов (выполняется кодирование данных), для их воспроизведения нужно выполнить декодирование. Программы, содержащие алгоритмы **код**ирования и **дек**одирования мультимедийных данных, называют **кодеками**. Кодеки входят в состав программ, обрабатывающих аудио- или видеофайлы, а также могут выпускаться как дополнительный набор программ, например **K-Lite Codec**.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ОБРАБОТКИ ОБЪЕКТОВ МУЛЬТИМЕДИА

Вы уже ознакомились с некоторыми программами для обработки мультимедийных данных – с аудио- и видеопроигрывателями. Более общая классификация таких программ по назначению представлена на рисунке 4.1.

Рассмотрим подробнее назначение этих программ и алгоритмы работы с ними.

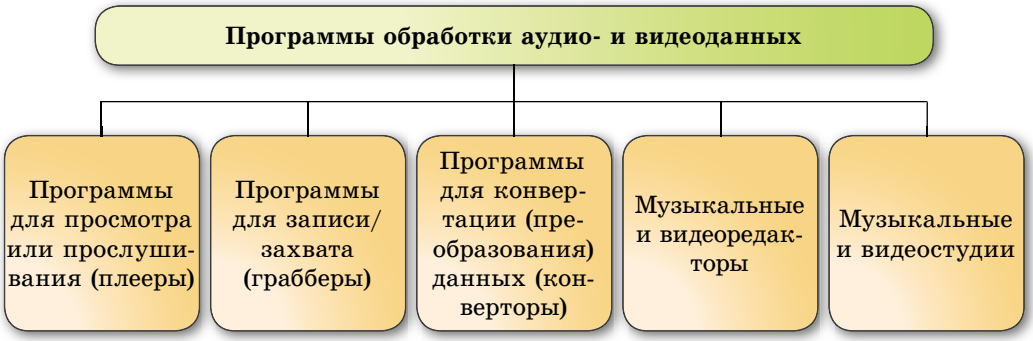


Рис. 4.1. Схема классификации программ для обработки аудио- и видеоданных

ЗАПИСЬ (ЗАХВАТ) АУДИО И ВИДЕО, СОЗДАНИЕ АУДИО-, ВИДЕОФРАГМЕНТОВ

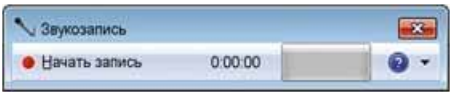


Рис. 4.2. Окно прогаммы **Звукозапись**

Для сохранения аудио- и видеоданных, вводимых с использованием устройств ввода/вывода мультимедийных данных, предназначены специальные ***программы записи (захвата) звука и видео***, которые еще называют **грабберами** (англ. *grabber* – тот, кто захватывает, хапуга, грабитель), или **рекордерами** (англ. *record* – запись). Данные будут сохранены в соответствующих файлах.

Используя эти программы, можно записать звуковой комментарий к презентации, видеоконференцию в **Skype**, веб-трансляцию спортивного соревнования и т. п.

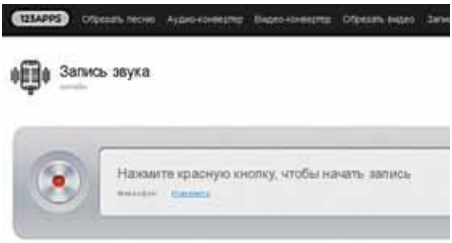


Рис. 4.3. Сервис Интернета для записи звука и видео

Примерами программ для захвата звука являются стандартная программа ОС **Windows Звукозапись** (рис. 4.2), **Audacity, Audiograbber, Streamripper, AML Easy Audio Recorder, Free Sound Recorder** и др. Можно также использовать интернет-сервисы для записи и сохранения звука, например:

http://vocaroo.com;
http://online-voice-recorder.com.

Для того чтобы записать звуковое сообщение с использованием микрофона, следует:

1. Подключить микрофон или воспользоваться встроенным (в мобильных компьютерах).
2. Запустить программу записи звука (например, ***Пуск*** ⇒ ***Все приложения*** ⇒ ***Стандартные – Windows*** ⇒ ***Звукозапись***).
3. Начать запись звука выбором соответствующей кнопки (например, **Начать запись** (рис. 4.3)).
4. Проговорить в микрофон необходимый текст.
5. Остановить запись выбором соответствующей кнопки (например, **Остановить запись**).
6. Сохранить звукозапись в аудиофайле в избранной вами папке.

В дальнейшем файл, который вы создали, можно открыть и прослушать в среде проигрывателя или редактора аудиофайлов.

Во время записи видео может выполняться запись видео и звука или только видео. Соответственно, к компьютеру следует подсоединить устройства ввода видео- и звуковых данных или только видеоданных.

Запись (захват) видео можно выполнять с веб-камеры, из окна проигрывателя видеофайлов или иной программы, с веб-страницы. Во всех случаях следует использовать соответствующую программу. Программы для записи видео – это **CamStudio, QIP Shot, Free Screen Video Recorder, Ezvid** и др. Например, для записи видео с использованием программы **QIP Shot** (рис. 4.4) следует:

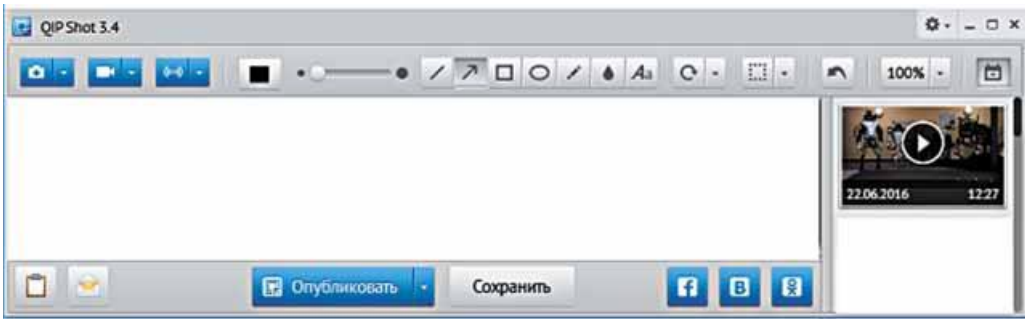


Рис. 4.4. Окно программы **QIP Shot**

1. Открыть окно программы, видео из которой нужно записать, или соответствующий сайт.
2. Запустить программу **QIP Shot**.
3. Выбрать режим записи в списке кнопки режимов съемки , например режим записи прямоугольной области экрана.
4. Указать, используя специальный курсор (рис. 4.5) программы, прямоугольную область, в которой будет выполнен захват видео.
5. Начать воспроизведение видео в программе или на сайте-источнике.
6. Включить запись видео выбором кнопки **Старт** (рис. 4.6).
7. Для завершения записи видео выбрать кнопку **Стоп**.
8. Для сохранения созданного видеофрагмента в файле выбрать кнопку **Сохранить**.
9. Указать папку, в которую будет сохранен файл, и имя файла.



Рис. 4.5



Рис. 4.6. Окно программы **QIP Shot** в режиме записи прямоугольной области экрана

Указанный алгоритм с небольшими изменениями может быть использован при записи видео с использованием и других программ.

СРЕДСТВА ПРЕОБРАЗОВАНИЯ АУДИО- И ВИДЕОФОРМАТОВ

При использовании аудио- и видеоданных часто возникает необходимость изменить формат файла, в который записаны эти данные. Например, если ваш мультимедийный проигрыватель не может воспроизвести файл определенного формата. Для таких случаев используют специальные программы – **конверторы**. Для конвертации аудиофайлов используют такие программы, как **HAMSTER Free Audio Converter**, **SoundConverter XRECODE II**, **TAudioConverter**, **AudioConverter Studio** и др.

Для конвертации аудиофайлов, например в программе **SoundConverter** (рис. 4.7), после запуска программы следует:

1. Выбрать кнопку Add File (англ. *Add File* – добавить файл) и в открывшемся окне выбрать файл для преобразования.
2. Выбрать кнопку **Параметры** и указать в списке **Format** формат файла, в который необходимо конвертировать файл.
3. Начать процесс конвертации выбором кнопки **Преобразовать**.

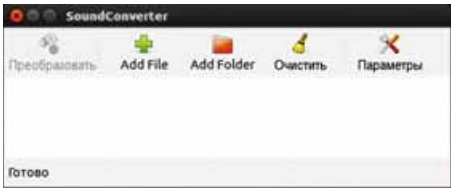


Рис. 4.7. Окно программы **SoundConverter**

Сконвертированный файл по умолчанию будет записан в ту же папку, где находился исходный файл. Установить другую папку для записи можно в окне, которое открывается после выбора кнопки **Параметры**.

Для конвертации видео существует много различных программ, которые распространяются по различным видам лицензий. Например, **HAMSTER Free Video Converter**, **AVS Video Converter**, **Any Video Converter Free**, **SUPER** и др.

Рассмотрим последовательность конвертации видеофайлов в программе **HAMSTER Free Video Converter**. После запуска программы следует:

1. Добавить файлы для конвертирования, выбрав кнопку **Добавить файлы**, выбрать в открывшемся окне необходимые файлы.
2. Выбрать кнопку **Далее**.
3. Указать формат файла, в который будете конвертировать, и при необходимости – значения свойств этого формата (рис. 4.8).
4. Выбрать кнопку **Преобразовать**.
5. Указать папку для записи конвертированного файла.



Рис. 4.8. Окно программы **HAMSTER Free Video Converter**

Также существуют веб-версии конверторов как аудио, так и видео, например, на сайтах *http://audio.online-convert.com*, *http://benderconverter.com*, *https://online-audio-converter.com*, *http://www.onlinevideoconverter.com*, *http://www.online-convert.com* и др.

Для тех, кто работает с Linux

Для инсталляции программ обработки аудио- и видеоданных, если они не были установлены при инсталляции ОС, можно использовать **Центр приложений Ubuntu**. Для этого в списке **Все программы** следует установить отображение программ работы со звуком и видео, выбрать нужную программу и кнопку **Установить** (рис. 4.9).

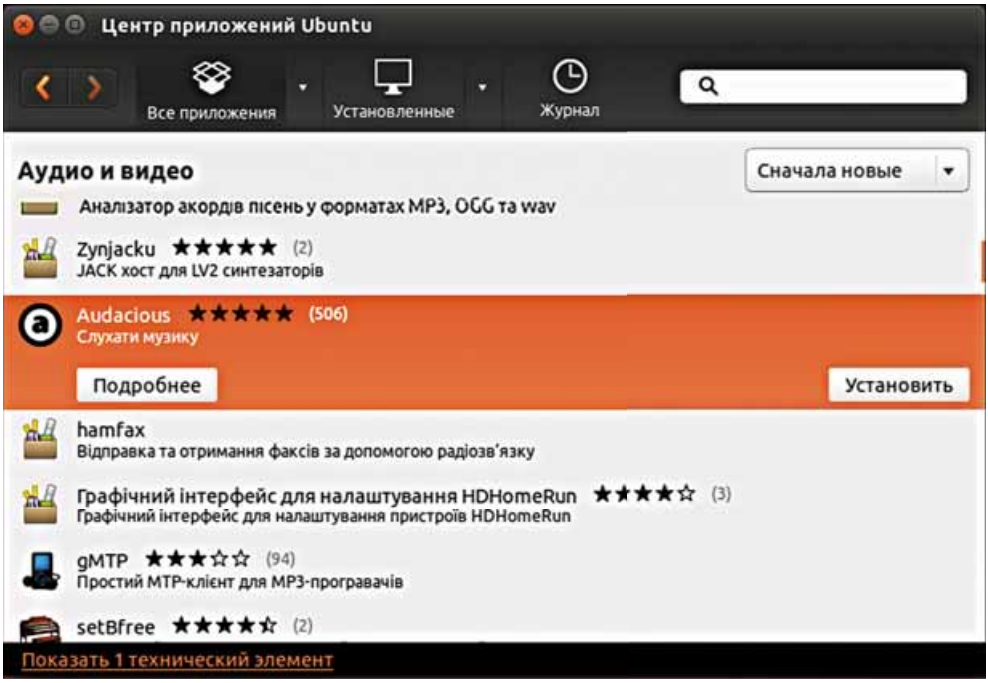


Рис. 4.9. Окно программы **Центра приложений Ubuntu**

Самое важное в этом пункте

Аудио- и видеоданные хранятся в файлах различных форматов. Файлы аудио имеют расширение имени – **wav**, **mid**, **flac**, **mp3** и др. Файлы видео имеют расширение имени – **mp4**, **wmv**, **mov**, **flv** и др.

Программы, содержащие алгоритмы **ко**дирования и **дe**кодирования мультимедийных данных, называют **кодеками**.

Программы, обрабатывающие видео и аудио, в зависимости от назначения, подразделяют на программы для просмотра или прослушивания **(плееры)**, программы для записи (захвата) звука и видео **(грабберы, рекордеры)**, программы для конвертации данных **(конверторы)**, музыкальные и видео**редакторы**, музыкальные и видео**студии**.

Дайте ответы на вопросы

1•. Какие типы (форматы) аудиофайлов вы знаете? Какие из них не предусматривают частичной потери данных?
2°. Какие типы (форматы) видеофайлов вы знаете?
3°. Что такое кодеки? Для чего они предназначены?
4°. Какие виды программ для обработки мультимедийных данных вы знаете?
5°. Для чего предназначены программы записи аудио и видео? Приведите примеры таких программ.
6•. Какова последовательность действий при записи аудиоданных?
7*. Чем отличаются действия при записи аудиоданных от действий при захвате видео?
8•. Из каких источников можно выполнить захват видео?
9•. Какова последовательность действий при конвертации аудио- и видеофайлов?

Выполните задания

1°. Запишите в собственном исполнении декламацию стихотворения Т. Шевченко «Думи мої, думи мої». **(Глава 4\Пункт 4.1\Думи мої. doc)** с использованием программы **Звукозапись** или другой по указанию учителя. Сохраните запись в вашей папке в файле с именем **задание 4.1.1.wav.**

2•. Используя интернет-ресурс для записи звука *http://online-voice-recorder.com*, запишите в собственном исполнении декламацию стихотворения В. Сосюры «Осінь» («Облітають квіти...».) **(Глава 4\Пункт 4.1\Облітають квіти.doc).** Сохраните запись в папке **Документы** в файле с именем **задание 4.1.2.mp3.**

3•. Выполните запись видео с веб-камеры вашего компьютера продолжительностью 30 с с использованием программы захвата видео **QIP Shot.** Сохраните запись в папке **Документы** в файле с именем **задание 4.1.3.mp4.**

4•. Выполните запись видео из окна видеопроигрывателя при воссоздании им файла **видео 2.avi (Глава 4\Пункт 4.1\видео 2.avi)** с использованием программы захвата видео **QIP Shot.** Сохраните запись в вашей папке в файле с именем **задание 4.1.4.wmv.**

5*. Выполните преобразование двух первых файлов формата **MP3** из папки, указанной учителем, например **Глава 4\Пункт 4.1\Аудио,** в формат **WMA,** используя один из конверторов звуковых файлов. Сохраните эти файлы в папке **Задание 4.1.5,** которую создайте в вашей папке. Сравните размеры начальных и полученных файлов. Чем вы объясните разницу в их размерах?

6*. Выполните преобразование двух первых файлов формата **MOV** из папки **Глава 4\Пункт 4.1\Видео** в формат **WMV,** используя один из конверторов видеофайлов. Сохраните эти файлы в папке **Задание 4.1.6,** которую создайте в папке **Документы.** Сравните размеры начальных и полученных файлов. Чем вы объясните разницу в их размерах?

4.2. РАЗРАБОТКА И СОЗДАНИЕ ВИДЕОФИЛЬМА

1. Какие форматы аудио- и видеофайлов вы знаете?
2. Какие виды программ для работы с мультимедийными данными вы знаете?
3. Какие действия над мультимедийными данными можно выполнить с использованием плееров, грабберов, конверторов?

ПРОГРАММЫ ДЛЯ РЕДАКТИРОВАНИЯ АУДИО- И ВИДЕОДАННЫХ

В предыдущем пункте мы не рассмотрели подробно два вида программ для работы с мультимедийными данными – музыкальные и видеоредакторы, а также музыкальные и видеостудии. Эти программы предназначены для редактирования звуковых и видеоданных. Используя их, можно вставлять, удалять, копировать, изменять продолжительность аудио- или видеофрагментов, объединять фрагменты, вставлять в видео текстовые и графические объекты и т. п.

Отличие между названными программами в том, что мультимедийные студии – это многофункциональные программы. Они имеют расширенный набор функций и используются в профессиональной деятельности на студиях звукозаписи, кино- и анимационных студиях для создания звукозаписей, аудио- и видеокомпозиций, анимации, кино- и видеофильмов, рекламных роликов, видеоклипов и т. п. Примерами программ – музыкальных студий являются **Linux MultiMedia Studio**, **Psycledelics Psycle**, **Steinberg Cubase**, **Cakewalk Sonar**, **Adobe Audition** и др. Программами видеостудий являются **Pinnacle STUDIO**, **Kino**, **Adobe Premiere Pro**, **Avid Media Composer** и др. Обычно программы-студии являются коммерческими программами.

Для создания и редактирования простых звуковых и видеофайлов используют различные мультимедийные редакторы. Например, для редактирования аудиоданных можно использовать **Audacity**, **Free Audio Editor**, **Wave Editor**, а для видеоданных – **Киностудию Windows**, **Open Shot**, **Virtual Dub**, **ZS4 Video Editor** и т. п.

КИНОСТУДИЯ WINDOWS

Киностудия Windows является новой версией редактора видеоданных **Windows Movie Maker**. Бесплатную версию программы пользователи **Windows 7** и **Windows 10** могут загрузить с сайта **Microsoft**. Для предыдущих версий ОС **Windows** можно использовать **Windows Movie Maker**, которая при установке ОС по умолчанию включается в состав стандартных программ.

Киностудия Windows предназначена для создания видеофильмов, которые могут содержать видеофрагменты, фотографии, звуковое сопровождение (*дикторский текст*), различные надписи (*титры*). Созданное видео можно опубликовать в Интернете или сохранить в файле одного из возможных форматов.

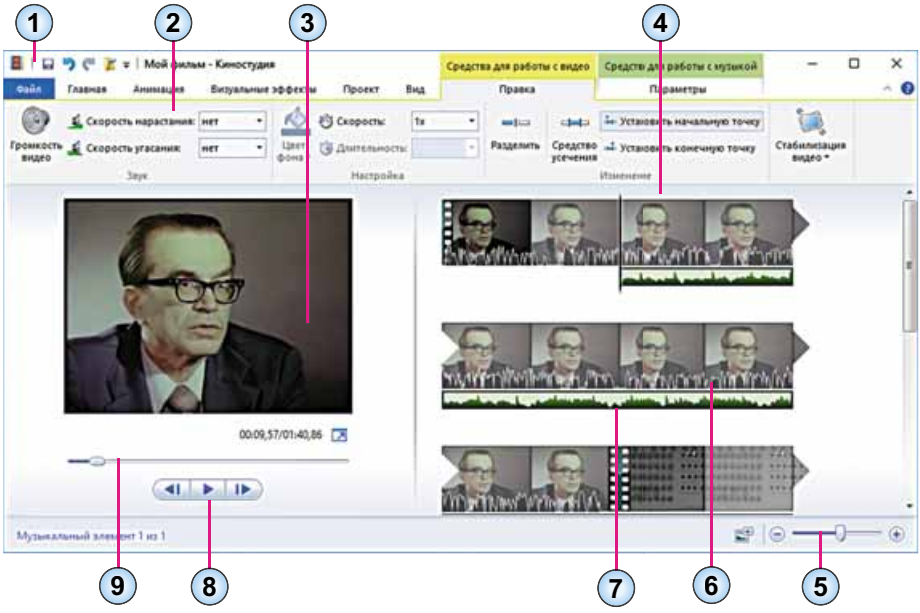
Общий вид окна программы **Киностудия Windows** представлен на рисунке 4.10.

ОБЩИЕ ПОНЯТИЯ О ВИДЕОФИЛЬМЕ

Под **видеофильмом**, как и **кинофильмом** (англ. *cine-film* – кинопленка, кинофильм), мы в основном понимаем произведение киноискусства. С распространением различных устройств для записи видеоданных создать видеофильм может любой пользователь. Такие видеофильмы, как правило, предназначены для просмотра дома или в небольших группах, в отличие от просмотра фильмов в кинотеатрах. Для небольших по размеру кино- и видеофильмов используют термин **клип** (англ. *clip* – сжимать, обрезать), или **видеоклип**.

Видеофильм состоит из **кадров** – отдельных фрагментов подвижного (видео) или неподвижного (фото) изображения. Кадры образуют видеоряд и размещаются в видеоредакторе на специальной видеодорожке. Видеофильм, как правило, содержит следующие объекты:

- **заголовок** – один или несколько текстовых объектов, которые появляются в начале видеофильма;
- **титры** – текстовые объекты, которые появляются или поверх видео и используются для объяснения изображения, или в конце видео и содержат сведения об авторах видеофильма;
- **звуковое сопровождение** может быть нескольких видов:
  - записанное вместе с видео;



1. Панель быстрого запуска
2. Лента
3. Окно просмотра
4. Область монтажа
5. Ползунок изменения масштаба области монтажа
6. Видеоряд (видеодорожка)
7. Звуковой ряд (звуковая дорожка)
8. Кнопки управления просмотром
9. Ползунок просмотра

Рис. 4.10. Окно программы **Киностудия Windows**

- дикторское – комментарий или объяснение к видео;
- музыкальное – музыкальное произведение, которое используется в качестве фонового во время просмотра видео;

- **визуальные эффекты** – используются для изменения всего или части изображения, например, эффект черно-белого изображения или эффект старой кинопленки;
- **эффекты перехода** – эффекты, используемые для плавного перехода от одного кадра к другому.

Визуальные эффекты и эффекты перехода отображаются, как правило, на видеодорожке. Звуковое сопровождение может размещаться на нескольких отдельных звуковых дорожках. В **Киностудии Windows** звук, записанный вместе с видео, отображается на видеодорожке.

При создании видеофильмов следует придерживаться определенной последовательности:

- ***на первом этапе*** разрабатывается сценарий – детальный план последовательности отдельных кадров фильма и размещения отдельных объектов. Чем детальнее будет сценарий, тем проще будет создавать видеофильм;
- ***на втором этапе*** выполняется подготовка нужных материалов – происходит съемка всех эпизодов согласно сценария, подбирается музыка, фото, рисунки, записывается звуковое сопровождение;
- ***на третьем этапе*** выполняется монтаж – обработка и структурирование отдельных кадров для получения целостного произведения – видеофильма. В ходе монтажа отснятый материал редактируется, уточняется сценарий, накладываются звук и титры – текстовые комментарии к кадрам.

РАЗРАБОТКА СЦЕНАРИЯ ВИДЕОФИЛЬМА

Рассмотрим процесс разработки сценария видеофильма на примере создания видеофильма «Безбумажная информатика» о реализации идей В.М. Глушкова об использовании компьютерных средств в наши дни. Он говорил, что на начало двадцать первого века основная масса информации будет храниться в безбумажном виде – в памяти ЭВМ.

Для реализации этих идей под руководством В.М. Глушкова была разработана серия ЭВМ **МИР** (***машина для инженерных расчетов***) – прообразы персональных компьютеров. Основная идея таких компьютеров – их широкое использование всеми слоями населения, независимо от уровня образования и возраста.

Учитывая цель использования нашего видеофрагмента – демонстрация на уроке информатики для иллюстрации сообщения об академике В.М. Глушкове, составим сценарий. Сценарий состоит из сценарных кадров – описания смысловых фрагментов видеофильма и событий, которые в них происходят.

1. Кадр 1. **Титульный:**
   - заголовок – *Безбумажная информатика*;
   - цвет фона – *светло-зелений*, цвет букв – *темно-зеленый*;
   - длительность показа – *7 с*;
   - звук – *общий фоновый звук из файла*;
   - визуальный эффект – *кинематографическое исчезание, слева*;

- эффект перехода на следующий кадр – *растворение, точечный эффект*.

2. Кадр 2. **Фотография В.М. Глушкова:**
   - фото из файла;
   - длительность показа – *6 с*;
   - звук – *дикторский текст, начинающийся на этом кадре и заканчивающийся на следующем*; длительность звучания – *определяется скоростью воспроизведения текста*;
   - визуальный эффект – *отсутствует*;
   - эффект перехода на следующий кадр – *растворение, точечный эффект*;
   - дикторский комментарий – *директор Института кибернетики Академии наук Украины академик В.М. Глушков одной из задач использования компьютерной техники считал переход к безбумажной информатике. Он говорил, что на начало XXI века основные объемы информации будут храниться в безбумажном виде – в памяти ЭВМ. И человек, который не будет уметь работать с компьютером и использовать эту информацию, будет подобен тому, кто в начале XX века не умел ни писать, ни читать.*
3. Кадр 3. **Видео с участием В.М. Глушкова:**
   - видео из файла;
   - длительность показа – *завершается одновременно с дикторским текстом*;
   - звук – *общий фоновый звук из файла*;
   - визуальный эффект – *отсутствует*.
4. Кадр 4. **Заголовок:**
   - текст – *Послушаем самого Виктора Михайловича Глушкова*;
   - цвет фона – *светло-зеленый*, цвет букв – *темно-зеленый*;
   - длительность показа – *7 с*;
   - звук – *общий фоновый звук из файла*;
   - визуальный эффект – *кинематографические исчезания, слева*;
   - эффект перехода на следующий кадр – *растворение, точечный эффект*.
5. Кадр 5. **Видеорассказы В.М. Глушкова о безбумажной информатике:**
   - видео из файла;
   - длительность показа – *48 с*;
   - звук – *звук из файла видео*;
   - визуальный эффект – *отсутствует*;
   - эффект перехода на следующий кадр – *растворение, точечный эффект*.
6. Кадр 6. **Фотография В.М. Глушкова вместе с разработчиками компьютера МИР-1:**
   - фото из файла;
   - длительность показа – *6 с*;
   - звук – *общий фоновый звук из файла*;
   - подпись – *Для того чтобы каждый человек мог работать с ЭВМ, нужно перейти к использованию малых компьютеров – В.М. Глушков выдвигает идею создания компьютеров МИР*;
   - цвет букв – *оранжевый*, цвет фона – *темно-зеленый*;

- визуальный эффект – *отсутствует*;
- эффект перехода на следующий кадр – *растворение, точечный эффект*.

7. Кадр 7. **Фотография компьютера МИР-1:**
   - фото из файла;
   - подпись – *Компьютер МИР имел небольшие размеры и удобный механизм ввода-вывода данных – электрическую печатную машинку*;
   - все другие значения свойств аналогичны кадру 6.
8. Кадр 8. **Фотография компьютера МИР-2:**
   - фото из файла;
   - подпись – *Компьютер МИР-2 имел дополнительную систему ввода данных – световое перо*;
   - все другие значения свойств аналогичны кадру 6.
9. Кадр 9. **Фотография компьютера МИР-3:**
   - фото из файла;
   - подпись – *Компьютер МИР-3*;
   - все другие значения свойств аналогичны кадру 6.
10. Кадр 10. **Заключительные титры:**
    - титры – *Школьная видеостудия «Будущее»*;
    - *оператор – Михайленко Степан*;
    - *автор текста – Коваленко Оксана*;
    - *монтаж – Иваненко Николай*;
    - цвет фона – *желто-зеленый*, цвет букв – *белый*;
    - длительность показа – *10 с*.

ПОДГОТОВКА МАТЕРИАЛОВ К ФИЛЬМУ

Используя материалы Интернета, печатные материалы, подберем файлы для использования в видеофильме. Их перечень представлен в таблице 4.2.

Таблица 4.2

Файлы для включения в видеофильм

Имя файла	*Папка размещения*	*Продолжительность, с*	*Номер кадра*
Глушков видео 0.*wmv*	Глава 4\Пункт 4.2\Видео	53	3
Глушков видео 1.*wmv*	Глава 4\Пункт 4.2\Видео	48	5
Glushkov.jpg	Глава 4\Пункт 4.2\Фото		2
Glushkov_MIR-1.*jpg*	Глава 4\Пункт 4.2\Фото		6
MIR-1.*jpg*	Глава 4\Пункт 4.2\Фото		7
MIR-2.*jpg*	Глава 4\Пункт 4.2\Фото		8
MIR-3.*jpg*	Глава 4\Пункт 4.2\Фото		9
Physical.wma	Глава 4\Пункт 4.2\Аудио	220	1–9, за исключением 5

СОЗДАНИЕ ВИДЕОФИЛЬМА

Чтобы создать видеофильм с использованием **Киностудии Windows**, следует:

1. Запустить программу.
2. Сохранить проект видеофильма – ***Файл*** ⇒ ***Сохранить проект***. По умолчанию программа сохраняет файл проекта в файле с расширением имени **wlmp**.
3. Вставить изображение, видео- и аудиоматериалы из файлов в последовательности, определенной сценарием, использовав для этого элементы управления группы **Добавление** вкладки **Главная** (рис. 4.11).

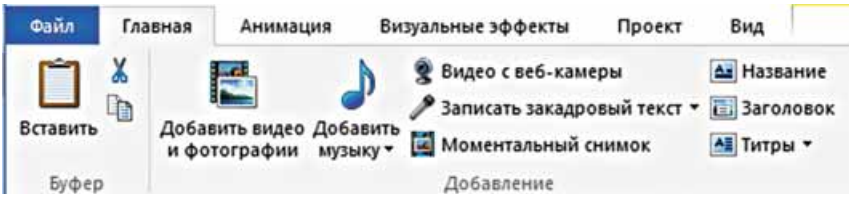


Рис. 4.11. Элементы управления группы **Добавление**

4. Добавить заголовки фильма, титры, подписи согласно сценарию, использовав для этого элементы управления группы **Добавление** вкладки **Главная**.
5. Отредактировать объекты фильма, используя элементы управления временных вкладок **Средства для работы с видео, Средства для работы с музыкой, Инструменты закадрового текста, Средства для работы с текстом.** При этом следует учитывать, что в области монтажа видео- и фотоматериалы вставляются в видеоряд (рис. 4.12, 2), звук из файла – в звуковой ряд (рис. 4.12, 4), звук дикторских комментариев –



1. Обозначение границы кадров
2. Видеоряд
3. Звуковая дорожка видео со звуком
4. Звуковой ряд звукового файла
5. Звуковой ряд дикторского комментария
6. Ряд текста

Рис. 4.12. Объекты **Области монтажа**

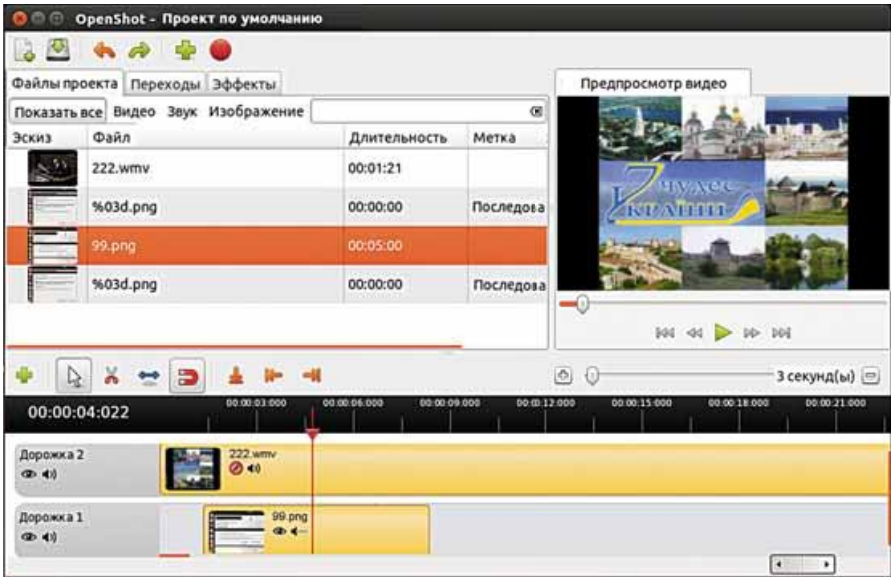
еще в один звуковой ряд (рис. 4.12, 5), а титры, заголовки и подписи – в отдельный ряд с текстом (рис. 4.12, 6). Если видео имеет звук, то звуковая дорожка отображается в нижней части видеоряда (рис. 4.12, 3).

6. Согласовать видеоряд с другими рядами по времени просмотра, по громкости звука и т. п., используя перемещения объектов по дорожке, перетаскиванием места начала и окончания объекта и т. п.
7. При необходимости вставить эффекты, которые будут накладываться на отдельные кадры (элементы управления вкладки **Визуальные эффекты**).
8. Вставить эффекты перехода между отдельными кадрами видеофильма (элементы управления вкладки **Анимация**).
9. Сохранить файл видеофильма – ***Файл*** ⇒ ***Сохранить фильм***.

В ходе монтажа видеофильма проект следует периодически сохранять. Если изменение видеофильма не завершено, то для его продолжения следует открыть сохраненный ранее файл проекта. Наличие сохраненного файла проекта также дает возможность пользователю создать несколько вариантов видеофильма с различными эффектами и настройками.

Для тех, кто работает с Linux

В **Linux** для редактирования видео можно использовать программу **OpenShot** (см. рис.). По своему интерфейсу и основным операциям она во многом подобна программе **Киностудия Windows**, описанной в этом пункте.



Окно программы **OpenShot**

Самое важное в этом пункте

Музыкальные и видеоредакторы, а также музыкальные и видеостудии – это программы, предназначенные для редактирования звуковых и

видеоданных. Используя их, можно вставлять, удалять, копировать, изменять продолжительность аудио- или видеофрагментов, объединять фрагменты, вставлять в видео текстовые и графические объекты и т. п.

Отличие между названными программами в том, что мультимедийные студии имеют расширенный набор функций и используются в профессиональной деятельности на студиях звукозаписи, кино- и анимационных студиях.

Киностудия Windows предназначена для создания видеофильмов, которые могут содержать видеофрагменты, фотографии, звуковое сопровождение (дикторский текст), различные надписи (титры). Созданное видео можно опубликовать в Интернете или сохранить в файле одного из возможных форматов.

При создании видеофильмов стоит придерживаться следующей последовательности:

- ***на первом этапе*** разрабатывается сценарий;
- ***на втором этапе*** выполняется подготовка аудио-, фото и видеоматериалов;
- ***на третьем этапе*** выполняется монтаж видеофильма.

Дайте ответы на вопросы

1°. Для чего предназначены музыкальные и видеоредакторы?
2•. В чем отличие между видеоредакторами и видеостудиями?
3°. Какие музыкальные редакторы вы знаете?
4°. Какие видеоредакторы вы знаете?
5°. Для чего предназначена **Киностудия Windows**?
6•. Опишите по рисунку 4.10 основные объекты окна программы **Киностудия Windows**.
7°. Для чего предназначена **Область монтажа** в программе **Киностудия Windows**?
8°. Какая последовательность создания видеофильма?
9•. Что такое сценарий видеофильма, для чего он создается?
10•. Как вставить в проект видеофильма видео или фото из файла?
11•. Как записать дикторский текст?
12*. В чем сущность согласования видео и звука? Как это выполнить в **Киностудии Windows**?

Выполните задания

1•. Составьте сценарий и создайте по нему в выбранном вами видеоредакторе видеофильм о Лесе Украинке к урокам украинской литературы. В видеофильме используйте фрагмент из фильма «Великие украинцы. Леся Украинка» и фотографии поэтессы и памятников Лесе Украинке из папки **Глава 4\Пункт 4.2\Украинка**. Подготовьте и включите в видеофильм титульный слайд и титры сопровождения. Самостоятельно подберите музыку, которая должна звучать в течение всего видеофильма. Сохраните видеофильм в вашей папке в файле с именем **задание 4.2.1.wmv**.
2•. Создайте в выбранном вами видеоредакторе видеофильм о выдающемся путешественнике Христофоре Колумбе, используя материал

учебника по Всемирной истории для 8-го класса и материалы, размещенные в папках **Глава 4\Пункт 4.2\Колумб**. Дикторский текст и музыкальное сопровождение подберите самостоятельно. Сохраните видеофильм в вашей папке в файле с именем **задание 4.2.2.wmv**.

3•. Разработайте сценарий и создайте в выбранном вами видеоредакторе видеофильм о выдающемся украинском писателе и кинорежиссере Александре Довженко для демонстрации на уроке украинской литературы. Используйте материалы учебника для 8-го класса по украинской литературе и материалы, размещенные в папках **Глава 4\Пункт 4.2\Довженко**. Дикторский текст и музыкальное сопровождение подберите самостоятельно. Сохраните видеофильм в папке **Документы** в файле с именем **задание 4.2.3.wmv**.

4*. Создайте в выбранном вами видеоредакторе видеофильм, который будет иллюстрировать один из физических процессов из курса физики 8-го класса. Сценарий и материалы подберите самостоятельно.

5•. Создайте в избранном вами видеоредакторе видеофильм о вашем классе или событиях в жизни школы. Сценарий и материалы подберите самостоятельно.

ПРАКТИЧЕСКАЯ РАБОТА № 6

«Создание видеоклипа. Добавление видеоэффектов, настройка временных параметров аудио- и видеоряда»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Создайте, используя программу видеоредактора, видеоклип с рассказом о термометрах длительностью до 2,5 минут.

2. Разработайте сценарий видеоклипа, предусмотрев:

- титульный лист с названием видеоклипа, например **Термометры**;
- обзор различных видов термометров, использовав для этого фотографии (4–6 фото) из папки **Глава 4\Практическая 6\Фото**. К каждому фото добавьте подписи с названием типа термометра (тип соответствует названию файла). Продолжительность демонстрации каждого из фото – не более 5 с;
- рассказ о термометре Галилея, использовав для этого 2–3 фотографии из папки **Глава 4\Практическая 6\Фото** и видео из папки **Глава 4\Практическая 6\Видео**. К фото добавьте подписи, а к видеофрагменту – дикторский текст. Для подготовки подписей и дикторского текста используйте материал из файла **Глава 4\Практическая 6\Термометр Галилея.doc**;
- конечные титры с собственной фамилией как сценариста и режиссера фильма;
- музыкальное сопровождение всех частей текста, за исключением тех, где используется дикторский текст. Для этого используйте файлы музыкальных произведений из папки **Глава 4\Практическая 6\Музыка**.

3. Сохраните видеоклип в вашей папке в файле с именем **практическая работа 6**.

4.3. РАЗМЕЩЕНИЕ МУЛЬТИМЕДИЙНЫХ ФАЙЛОВ В ИНТЕРНЕТЕ

1. Что такое аккаунт? Как создать аккаунт на почтовом сервере?
2. Как загрузить созданный ранее проект в **Киностудии Windows**?
3. Как добавить музыку к фильму в редакторе видео?

РАЗМЕЩЕНИЕ ВИДЕОФАЙЛОВ В СЕРВИСЕ YOUTUBE

Одним из распространенных сервисов для размещения видеофайлов в Интернете является сервис YouTube (англ. *You Tube* – твоя труба (трубка)). Размещать материалы и предоставлять доступ к ним на этом сервисе может любой пользователь, имеющий аккаунт почтовой службы **Google** – **Gmail**.

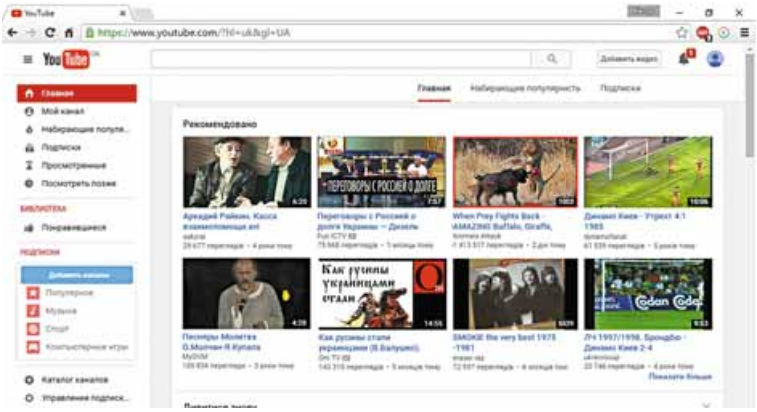


Рис. 4.13. Главная страница сайта **YouTube**

Для размещения мультимедийных материалов в **YouTube** следует:

1. Открыть в браузере главную страницу сервиса *https://www.youtube.com* – (рис. 4.13).
2. Выбрать в левой части окна ссылки **Мой канал**.
3. Ввести в окне логин и пароль почты **Gmail**.
4. Ввести имя и фамилию, под которыми ваши видеоматериалы будут размещаться в **YouTube** (по умолчанию будут использованы данные из аккаунта **Gmail**) (рис. 4.14).
5. Ознакомиться с условиями использования сервиса по соответствующей ссылке.
6. Выбрать кнопку **Создать канал**.
7. Выбрать кнопку **Загрузить** в верхней правой части окна.



Рис. 4.14. Диалоговое окно создания собственного канала в **YouTube**

8. Выбрать в списке кнопки **Конфиденциальность** (рис. 4.15) один из вариантов доступа к просмотру вашего файла: *Открытый доступ*, *По ссылке* или *Ограниченный доступ*.
9. Выбрать кнопку и выбрать файл для размещения в **YouTube**.
10. Дождаться завершения загрузки (или во время загрузки) и ввести значения свойств вашего видео в окне настроек (рис. 4.16):
    - название видео (рис. 4.16, 3);
    - описание видео – коротко описать, о чем ваше видео (рис. 4.16, 4);
    - ключевые слова (теги) для включения вашего видео в систему быстрого поиска пользователями по указанным тегам (рис. 4.16, 8);
    - адреса ваших друзей или знакомых для получения ими сообщения о размещении вами видео с адресом в **YouTube** (рис. 4.16, 7) – в случае выбора уровня конфиденциальности *Ограниченный доступ*.
11. Выбрать кнопку **Опубликовать**.

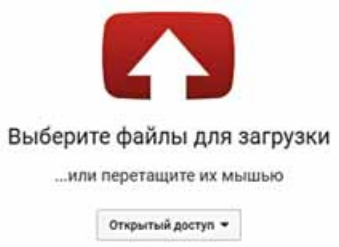


Рис. 4.15. Окно загрузки файла

Конфиденциальный (лат. *confidentia* – доверие) – не подлежащий разглашению, ограниченный в доступе.



1. Окно эскиза видео
2. Индикатор загрузки файла
3. Поле названия видео
4. Поле описания видео
5. Кнопка **Опубликовать**
6. Кнопка со списком **Конфиденциальность**
7. Поле адресатов видео
8. Поле ключевых слов (тегов)

Рис. 4.16. Окно настройки параметров видео

После загрузки видеофайла его можно просмотреть, а также выполнить редактирование, используя кнопки, появляющиеся в режиме просмотра под экраном (рис. 4.17). Можно изменить:

- название и описание (кнопка **Информация и настройки**);

- яркость, контрастность, насыщенность цветов, скорость просмотра и т. п. (кнопка **Улучшить видео**);
- добавить звук (кнопка **Аудио**);
- добавить титры (кнопка **Субтитры**)

и т. п.



Рис. 4.17. Кнопки средств редактирования видео

В дальнейшем пользователь может загрузить другие файлы, используя созданный видеоканал на **YouTube**.

Для размещения материалов в Интернете можно также использовать другие сервисы, например **Vimeo** (*https://vimeo.com*), **dailymotion** (*http://www.dailymotion.com*), **Видео ukrhomenet** (*http://video.ukrhome.net*) или социальные сети, например **Facebook**.

ИСПОЛЬЗОВАНИЕ КИНОСТУДИИ WINDOWS ДЛЯ РАЗМЕЩЕНИЯ ВИДЕО В ИНТЕРНЕТЕ

После разработки проекта видеофильма в программе **Киностудия Windows** можно не хранить файл видео во внешней памяти компьютера, а разместить его в Интернете. Для этого нужно открыть проект созданного видеофильма и выполнить следующий алгоритм:

1. Выполнить ***Файл*** ⇒ ***Опубликовать фильм***.
2. Выбрать из списка сервисов (рис. 4.18) тот, на сайте которого будет размещен видеофайл, например **YouTube**.
3. Ввести в поля диалогового окна логин и пароль доступа к вашему видеоканалу на **YouTube** (обычно совпадает с аккаунтом **Gmail**).
4. Ввести в поля диалогового окна значения свойств видеофильма: *название, описание, теги, вид разрешения на использование, категорию видеофильма*.
5. Выбрать кнопку **Опубликовать**.
6. Дождаться завершения сохранения и опубликования фильма.
7. Просмотреть видеофильм онлайн, выбрав соответствующую кнопку.
8. Закрыть окно браузера.



Рис. 4.18. Список вариантов опубликования фильма

По подобному алгоритму происходит опубликование фильма и на других сервисах. Обязательным условием размещения мультимедийных данных является создание аккаунта на сайте соответствующего сервиса.

РАЗМЕЩЕНИЕ АУДИОФАЙЛОВ В ИНТЕРНЕТЕ

Подавляющее большинство сервисов для размещения аудиофайлов предназначено для тех, кто создает собственные файлы с различными музыкальными произведениями. Еще одним направлением размещения аудиофайлов является создание аудиокниг и различных обзоров по определенным темам: политическим, экономическим, техническим и т. п.

Подкаст (англ. ***ipod*** – медиаплеер корпорации Apple; *broad* ***casting*** – эфирное вещание) – медиафайл или серия таких файлов, создаваемых произвольными пользователями (*подкастерами*) и распространяемых с использованием Интернета.

Указанные аудиофайлы называют **подкастами**. В дальнейшем термин «подкаст» распространился и на видеофайлы, размещаемые пользователями в Интернете.

Для размещения аудиофайлов в Интернете существуют различные сервисы, такие как: **Канал общественного подкастинга** (*http://cpod.co/*), **Украинский подкаст-терминал** (*http://podcaster.org.ua*), **Best Podcast Directory** (*http://www.podcast.com*), страница медиакомпании BBC для украинцев, изучающих английский (*http://www.bbc.co.uk/ukrainian/learningenglish*) и т. п.

Для обмена аудиофайлами с друзьями и знакомыми можно использовать сервис **Google Диск**. Для размещения файлов на **Google Диске** нужно иметь аккаунт **Google** и выполнить следующий алгоритм:

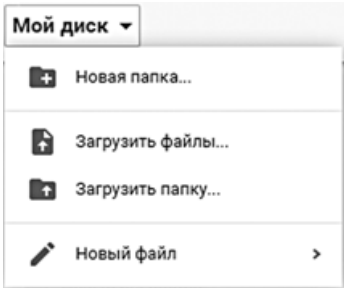


Рис. 4.19. Список кнопки **Мой диск**

1. Запустить браузер.
2. Открыть страницу **Google**, выбрать кнопку **Войти** и ввести логин и пароль аккаунта **Google**.
3. Выбрать кнопку **Приложения Google** и открыть список сервисов **Google**.
4. Выбрать в списке сервисов .
5. Открыть список кнопки **Мой диск** и выбрать команду **Загрузить файлы** (рис. 4.19).
6. Выбрать файл, который необходимо загрузить, и нажать кнопку **Открыть**.

В дальнейшем можно настроить доступ другим пользователям к вашим аудиофайлам. Например, после выбора кнопки **Совместный доступ** откроется окно настроек доступа к файлу другим пользователям (рис. 4.20). В поле **Люди** можно ввести электронные адреса ваших друзей, которым будет автоматически отправлено электронное письмо с адресом аудиофайла и разрешено его прослушивание.



Рис. 4.20. Окно настроек доступа к файлу

Выбор ссылки **Включить доступ по ссылке** устанавливает доступ к файлу тех пользователей, которые получат от вас ссылку с адресом размещения файла в Интернете. При этом данный адрес автоматически копируется в буфер обмена и может быть вложен в ваше электронное письмо друзьям, если вы решите направить отдельные письма, не заполняя поле **Люди**.

Для тех, кто работает с Linux

В **Linux** для размещения видео в Интернете можно использовать редактор видеофильмов **OpenShot**. После запуска программы следует:

1. Выполнить ***Файл*** ⇒ ***Загрузить видео на сервис***.
2. Указать видеофайл, который будет опубликован в Интернете, для чего воспользоваться элементом управления **Видеофайл** (рис. 4.21).
3. Выбрать сервис, на котором будет размещен видеофильм (элемент управления **Загрузить в**), например **YouTube**.
4. Ввести в соответствующие поля окна **Загрузить видео** значения свойств видеофильма: *название* (*заголовок*) и *описание*.
5. Ввести логин (*имя пользователя*) и *пароль* доступа к сервису.
6. Выбрать кнопку **Загрузить видео**.

Все остальные действия по размещению видеофайла аналогичны рассмотренным выше.



Рис. 4.21. Окно **Загрузить видео**

Самое важное в этом пункте

В Интернете существует множество сайтов, предоставляющих услуги по размещению мультимедийных файлов. Обычно для размещения файлов на сайтах необходимо иметь соответствующий аккаунт.

Одним из распространенных сервисов для размещения видеофильмов является **YouTube** (*https://www.youtube.com*). Для загрузки фильмов на сайт **YouTube** необходимо создать собственный канал, который в дальнейшем будет содержать подборку видеофайлов пользователя. Пользователь может управлять доступом к просмотру каждого из видеофайлов, устанавливая определенный уровень конфиденциальности.

Значительная часть редакторов видео имеют средства для размещения видеофильмов в Интернете, например такие программы, как **Киностудия Windows** и **OpenShot**.

Мультимедийные файлы, предназначенные для размещения в Интернете, называют **подкастами**, а тех, кто создает такие файлы, **подкастерами**.

Дайте ответы на вопросы

1°. Какие вы знаете сервисы для размещения мультимедийных файлов в Интернете?

2•. Как создать канал для размещения видеофайлов на сайте **YouTube**?

3°. Какие значения свойств видео могут быть установлены пользователем при загрузке файла на сайт **YouTube**?

4°. Какие уровни конфиденциальности можно установить для просмотра загруженного на **YouTube** файла?

5•. Какие действия следует выполнить для добавления музыкального сопровождения к загруженному на **YouTube** файлу?

6•. Какие действия нужно выполнить для размещения видеофайла на одном из сервисов Интернета непосредственно из программы видеоредактора, например, из **Киностудии Windows**?

7•. Какова последовательность действий для размещения аудиофайлов на **Диске Google**?

8•. Что такое подкаст? Кто такой подкастер?

9•. Как предоставить доступ к файлу другим пользователям с использованием **Диске Google**?

Выполните задания

1°. Просмотрите видеофильм по адресу *https://youtu.be/NcrOD9RysUo*. Определите продолжительность демонстрации.

2•. Разместите на **YouTube** предложенный учителем файл, например **Глава 4\Пункт 4.3\Видео\Сейм, старое речище.mov**. Добавьте описание, теги и музыкальное сопровождение. Отправьте учителю адрес размещенного файла.

3•. Используя проект видеофильма об А. Довженко, созданного на прошлых занятиях, разместите на **YouTube** этот фильм, непосредственно из программы **Киностудия Windows**.

4•. Разместите на **YouTube** предложенный учителем файл, например **Глава 4\Пункт 4.3\Видео\Улитка.mov**. Добавьте описание, теги. Найдите в Интернете сведения об улитках и добавьте их к видеофильму в виде титров.

5•. Разместите на **Диске Google** предложенный учителем файл, например **Раздел 4\Пункт 4.3\Аудио\Щедривочка.wma**. Предоставьте доступ к этому файлу тем пользователям, которые имеют ссылку на него. Отправьте по электронной почте адрес файла в Интернете учителю и одному из ваших друзей.

ПРАКТИЧЕСКАЯ РАБОТА № 7
«Размещение аудио- и видеоматериалов в Интернете»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Разместите на **YouTube** предложенный учителем файл, например **Глава 4\Практическая 7\Видео\Насекомые.wmv**.

2. Установите следующие значения его свойств:
- название – *Насекомые*;
- описание – *Насекомые центральной части Украины*;
- теги – *природа, насекомые, жук-олень*;
- уровень конфиденциальности – *ограниченный доступ*;
- эскиз видео – *жук-олень*;
- электронные адреса доступа – *адрес учителя*;
- титры – *с 00.01.0 до 00.09.0* текст *Жук-олень* (*Lucanus cervus*) – *самый известный вид из семейства рогачей*;
- музыка – вид **Кантри и фольк**, название **As We Go**.

3. Разместите аудиофайл, предложенный учителем, например **Глава 4\Практическая 7\Аудио\Музыка.mp3**, на **Диске Google** и предоставьте доступ к его прослушиванию учителю и одному из одноклассников.

Глава 5. Основы событийно- и объектно-ориентированного программирования

В этой главе вы узнаете:

- о языках программирования и их составляющих
- о среде разработки проектов **Lazarus**
- о создании проектов в среде **Lazarus** на языке **Object Pascal**, их редактировании и исполнении
- о компонентах: форме, кнопке, метке; их свойствах и событиях, которые могут с ними происходить
- об обработчиках событий
- об изменении значений свойств компонентов перед исполнением проектов и во время их исполнения
- об окне сообщений

5.1. КОМПЬЮТЕРНЫЕ ПРОГРАММЫ И ЯЗЫКИ ПРОГРАММИРОВАНИЯ

1. Что такое алгоритм? Что такое система команд исполнителя алгоритма? В чем суть формального исполнения алгоритма?
2. Что такое компьютерная программа?
3. Какой язык программирования вы изучали в предыдущих классах? Вспомните несколько команд этого языка программирования.

КОМПЬЮТЕРНЫЕ ПРОГРАММЫ

Вы уже знаете, что компьютер работает под управлением программного обеспечения, которое состоит из компьютерных программ различного назначения. Работая с компьютером в школе и дома, вы использовали текстовый процессор, графический редактор, программы-архиваторы, табличный процессор, редактор компьютерных презентаций, учебные и контролирующие программы, игровые программы и многие другие.

Вы также знаете, что **компьютерная программа** – это алгоритм обработки данных, записанный на специальном языке и предназначенный для исполнения компьютером.

В процессе работы программа обрабатывает **данные**.

Данные, которые передаются в программу с некоторых устройств (например, с клавиатуры или с датчика температуры), из другой программы или другого источника (например, из текстового файла), называют **входными (начальными) данными**. Некоторые программы работают без входных данных.

При исполнении программы создаются и обрабатываются и другие данные, которые называют **промежуточными данными**.

Если целью исполнения программы является получение определенных данных, то эти данные называют **выходными (результирующими) данными**.

Большинство современных программ в процессе работы предоставляют пользователю определенный набор средств для его взаимодействия с программой и устройствами. К этим средствам относятся средства управления (кнопки, меню и др.), средства ввода данных (поля, счетчики и др.), средства вывода данных (метки, поля и др.) и т. п. Совокупность таких средств, а также методов их использования образует **интерфейс пользователя**.

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Составляя алгоритмы, предназначенные для исполнения человеком, используют язык общения людей: украинский, русский, английский, немецкий и др.

Но для алгоритмов, которые должен исполнить компьютер, язык общения людей сложен, имеет неоднозначные трактовки (например, слова-омонимы). Поэтому для записи алгоритмов, предназначенных для исполнения компьютером, разрабатывают и используют специальные языки – **языки программирования**.

Язык, который используют для записи алгоритмов, предназначенных для исполнения компьютером, называют **языком программирования**.

За последние 70 лет создано около трех тысяч различных языков программирования. Некоторые из них уже не используют, для некоторых постоянно появляются новые версии, удобные для составления программ и имеющие большие возможности. Постоянно создаются новые языки программирования.

Некоторые языки программирования используются для составления программ для решения задач в различных областях науки, техники, в сфере быта и др. А некоторые созданы для составления программ для решения специального круга задач.

Языки программирования являются примерами так называемых **искусственных языков**. Другими примерами искусственных языков являются, например, языки записи химических реакций, язык записи решения математических уравнений и неравенств, язык записи почтовых индексов на конвертах и открытках, язык записи ходов шахматной партии и др.

Каждый язык программирования имеет следующие составляющие:

1. **Алфавит** – множество символов, из которых можно создавать слова и предложения этого языка.
2. **Словарь** – набор слов, которые используются в этом языке.
3. **Синтаксис** – правила составления и записи речевых конструкций: не словарных слов и предложений.
4. **Семантику** – установленное однозначное толкование языковых конструкций, правил их исполнения.

В шестом–седьмом классе вы изучали язык программирования **Scratch**. Вспомним, что в алфавит этого языка программирования входят цифры, буквы русского алфавита, английского алфавита, символы :, =, +, *, < и др. В его словарь входят, например, слова **повторить**, **идти**, **если**, **оттолкнуться**. Согласно синтаксических правил этого языка программирования команда цикла с предусловием должна содержать слова **повторять пока** и именно в таком порядке. А в соответствии с семантикой этого языка программирования компьютер однозначно исполнит команду **идти на 10 шагов** и переместит исполнителя на 10 шагов в направлении его движения.

Использование символов, не входящих в алфавит языка программирования, неправильное написание словарных слов, нарушение синтаксических правил приводят к невозможности исполнения компьютером соответствующей команды. Такие нарушения называют **синтаксическими ошибками**.

В восьмом и девятом классе вы будете изучать язык программирования **Object Pascal** (англ. *Object Pascal* – объектный Паскаль). Его используют ученики и студенты для изучения основных принципов программирования, а также профессиональные программисты для создания разнообразных программ.

Интересные факты из истории

Язык программирования **Pascal** создала в 1970 г. группа сотрудников Института информатики Швейцарской высшей технической школы под руководством профессора Никлауса Вирта (род. в 1934 г., рис. 5.1). Назвали его в честь выдающегося французского математика, физика, инженера, литератора, философа Блеза Паскаля.

Сначала язык программирования **Pascal** создавался специально для обучения студентов основным принципам и методам процедурного программирования. Но впоследствии выяснилось, что этот язык программирования достаточно

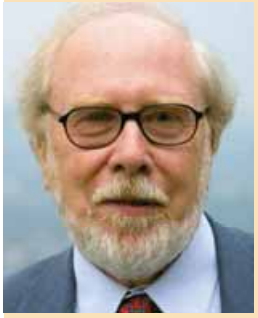


Рис. 5.1. Никлаус Вирт

удобен, и возникла идея использовать его не только для обучения, но и для создания программ для решения практических задач с использованием компьютера. Так появились первые программы-компиляторы для автоматического перевода программ, написанных на языке программирования **Pascal**, в программы, которые может исполнить процессор компьютера, а также среда разработки **Turbo Pascal**, которые со временем получили широкую известность.

С годами язык программирования **Pascal** развивался и совершенствовался. Новые версии этого языка, которые уже имели элементы **объектно-ориентированного** (ориентированного на объекты и их свойства) языка программирования, появились во второй половине 80-х годов прошлого века. Одна из версий этого языка получила название **Object Pascal**. В начале 2000-х годов очередная новая версия **Object Pascal** получила название **Delphi**.

СРЕДА РАЗРАБОТКИ ПРОГРАММ LAZARUS

Для создания, редактирования и отладки программ на языке программирования **Object Pascal** мы будем использовать **среду разработки программ Lazarus.**

Эта среда состоит из:

- **редактора исходного кода** для ввода и редактирования текста программ;
- **компилятора** для перевода программы с языка программирования **Object Pascal** на язык команд, которые может исполнить процессор компьютера;
- **средств отладки программы** для поиска в ней ошибок;
- **справочной системы**

и других компонентов.

Это среда разработки предоставляет возможность использовать большую **библиотеку визуальных компонентов** (англ. ***Lazarus Component Library* (*LCL*)** – библиотека компонентов **Lazarus).** Это уже известные вам компоненты: кнопки, поля, метки, флажки, переключатели, счетчики, списки, полосы прокрутки и др. При размещении этих компонентов на форме автоматически создается соответствующий фрагмент программы на языке **Object Pascal** и его не нужно вводить в текст программы вручную, что значительно уменьшает, упрощает и ускоряет работу по созданию программ. Такой способ разработки программ называют **визуальным.**

Программы, созданные в среде **Lazarus**, называют **проектами.** На носитель данных они записываются в нескольких файлах.

Открыть окно среды **Lazarus** можно:

- используя команду **Lazarus** меню ***Пуск*** ⇒ ***Программы*** ⇒ ***Lazarus***;
- используя значок  на **Рабочем столе.**

В результате открывается окно среды **Lazarus** (рис. 5.2), на котором размещено **Главное меню**, панели инструментов, вкладки с компонентами, а также несколько подчиненных окон, среди которых:

- окно **Инспектор объектов**, в котором можно для каждого компонента просматривать и изменять значения его свойств, начинать создание обработчиков событий и др.;

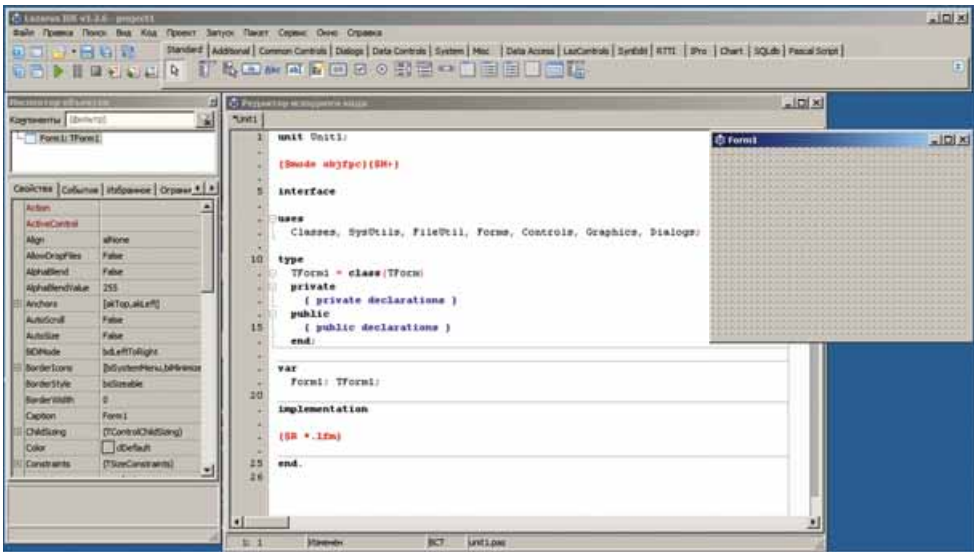


Рис. 5.2. Окно среды **Lazarus** и подчиненные окна

- окно **Редактор исходного кода** с вкладками, на которых размещаются фрагменты программы на языке **Object Pascal;**
- окно **формы.**

Форма – это объект, на котором можно разместить различные компоненты, например, кнопки, поля, надписи, меню, полосы прокрутки и др. В ходе исполнения проекта форма открывается как окно приложения, на котором отображаются все объекты, расположенные на форме.

Форма имеет **свойства**, которые отображаются на вкладке **Свойства** окна **Инспектор объектов**. В левой части этой вкладки находится список свойств, а в правой – их значения. Значения некоторых свойств формы необходимо вводить в соответствующие поля, а некоторые можно как вводить в поле, так и выбирать из списка.

Приведем примеры некоторых из этих свойств:

Name (англ. *name* – имя) – его значение определяет имя формы как объекта;

Caption (англ. *caption* – заголовок) – его значение определяет текст в строке заголовка формы;

Color (англ. *color* – цвет) – его значение определяет цвет фона формы;

Height (англ. *height* – высота) – его значение определяет высоту окна в пикселях (**пиксель** – это точка графического изображения на экране);

Width (англ. *width* – ширина) – его значение определяет ширину окна в пикселях;

Top (англ. *top* – верх) – его значение определяет расстояние (в пикселях) от верхней границы окна до верхней границы экрана;

Left (англ. *left* – левый) – его значение определяет расстояние (в пикселях) от левой границы окна к левой границе экрана.

Отметим, что значения свойств **Top** и **Left** можно изменить перетаскиванием формы по экрану, а значения свойств **Height** и **Width** – изменением размеров формы, используя маркеры, расположенные на ее границах.

Как мы уже отмечали, современные среды визуальной разработки проектов, одним из которых является **Lazarus**, частично автоматизируют процесс их создания, тем самым облегчая работу как профессионалам, так и начинающим. Уже даже при открытии нового проекта, когда еще не было введено ни одной строчки текста программы, **Lazarus** автоматически создает (генерирует) определенный исходный текст проекта. Этого текста проекта достаточно, чтобы запустить его на исполнение.

Перед началом исполнения проект рекомендуется сохранить на носителе. Как мы уже отмечали, современные проекты сохраняются на носителе данных не в одном файле, а в нескольких. Поэтому целесообразно для каждого нового проекта создавать отдельную папку. Сохранить проект можно, используя команду **Сохранить все** меню **Файл**, или кнопку **Сохранить все**  панели инструментов, или команду **Сохранить проект** меню **Проект**.

Система последовательно предлагает сохранить два файла проекта:

- файл с именем **Project1** (англ. *project* – проект) и с расширением **lpi**;
- файл с именем **Unit1** (англ. *unit* – блок) и с расширением имени **pas**.

Расширение имен файлов менять нельзя, а имена файлов можно изменить.

Все другие файлы проекта автоматически сохранятся в созданной папке, например, **Мой первый проект** (рис. 5.3).

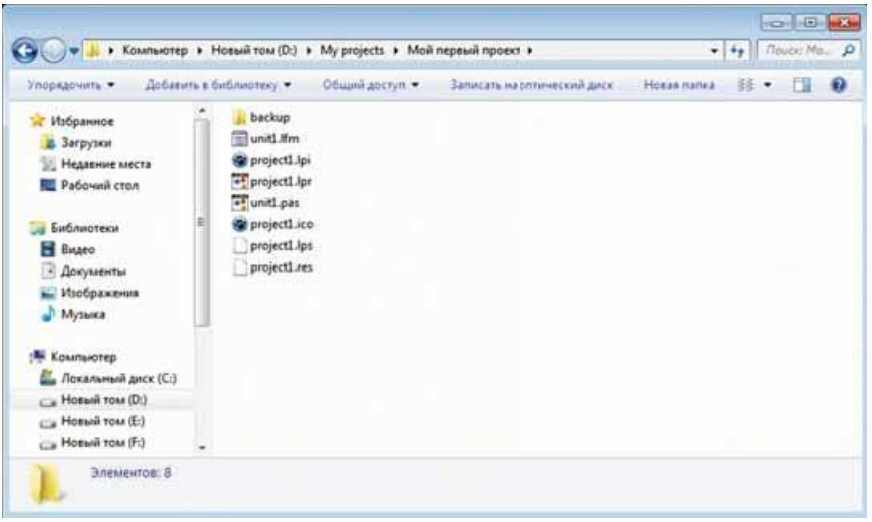


Рис. 5.3. Папка с файлами проекта

Если сохраненный проект изменить, то при сохранении его новой версии в ту же папку без изменения имен его файлов никакие дополнительные окна открываться не будут.

Уже сохраненный проект можно сохранить в другой папке, имена его файлов можно изменить. Для этого целесообразно использовать команду **Сохранить проект как** меню **Проект**.

Чтобы открыть проект, сохраненный ранее на носителе, следует выполнить команду **Открыть** из меню **Файл**, или выбрать кнопку **Открыть** 

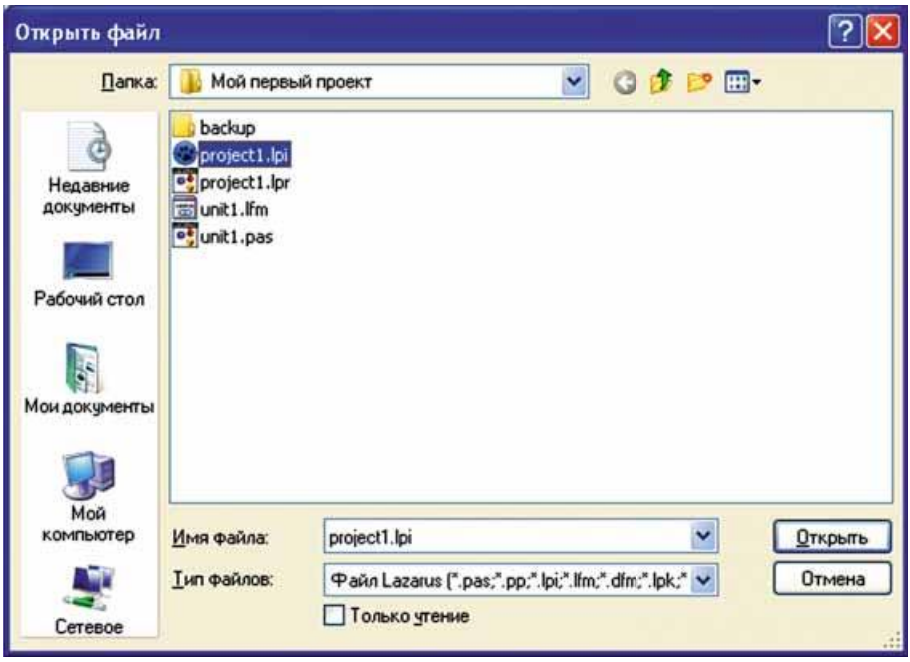


Рис. 5.4. Окно открытия проекта

на панели инструментов, или выполнить команду **Открыть проект** из меню **Проект.** После этого в открывшемся окне выбрать файл с расширением имени **lpi**, а далее выбрать кнопку **Открыть** (рис. 5.4).

Иногда при запуске среды **Lazarus** открывается последний сохраненный проект. Закрыть его можно командой **Закрыть проект** из меню **Проект.** После выполнения этой команды открывается окно **Мастер создания проекта** (рис. 5.5). В этом окне можно выбрать кнопку **Новый проект** для начала работы над новым проектом, пользуясь списком **Открыть недавний проект**, можно открыть один из последних проектов, которые открывались в среде, можно открыть любой проект, сохраненный на носителе.

После выбора кнопки **Новый проект** открывается окно **Создать новый проект** (рис. 5.6), в котором следует в списке выбрать первую строку **Приложение**, после чего выбрать кнопку **ОК.**

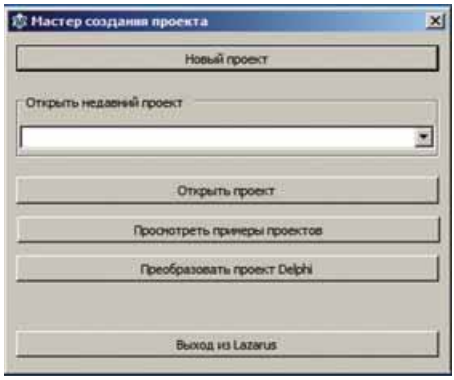


Рис. 5.5. Окно **Мастер создания проекта**

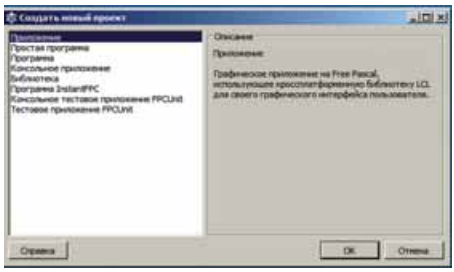


Рис. 5.6. Окно **Создать новый проект**

ИСПОЛНЕНИЕ ПРОЕКТА

Приступить к исполнению проекта можно, выполнив команду **Запустить** из меню **Запуск**, или выбрав кнопку **Запустить** на панели инструментов, или нажав клавишу **F9**.

При запуске проекта на исполнение начинает свою работу **программа-компилятор**. Прежде всего эта программа анализирует текст проекта на

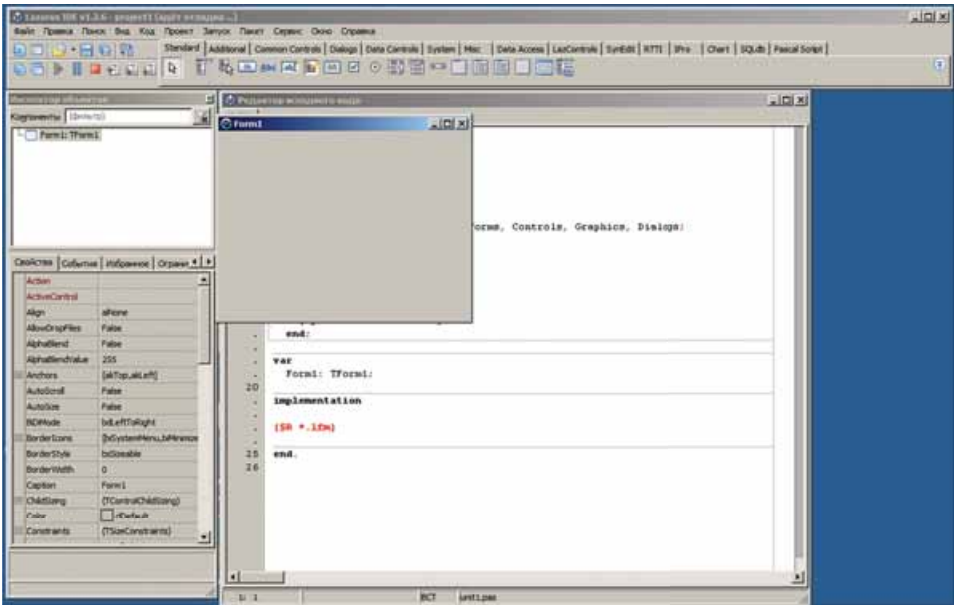


Рис. 5.7. Проект, который создается по умолчанию, и результат его исполнения

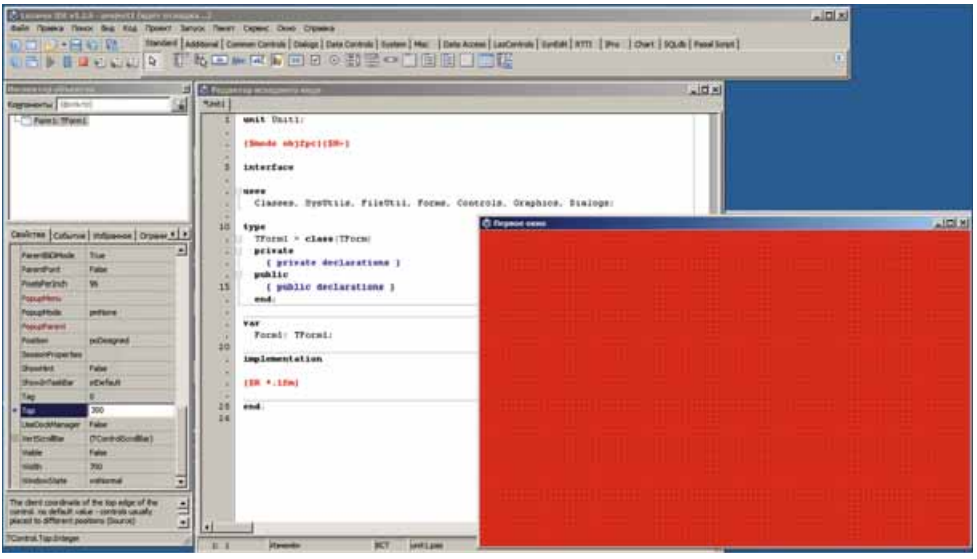


Рис. 5.8. Результат исполнения измененного проекта, созданного по умолчанию

наличие синтаксических ошибок. При наличии таких ошибок эта программа прерывает свою работу и указывает на место первой из них. Необходимо исправить ошибку и снова запустить проект на исполнение. Если текст проекта не содержит синтаксических ошибок, то программа-компилятор переводит текст проекта с языка **Object Pascal** на язык команд, которые может исполнить процессор компьютера, и передает уже переведенный проект процессору на исполнение.

В ходе исполнения проекта, который автоматически создается при открытии нового проекта, на экране открывается окно, значения свойств которого (цвет, размеры, положение на экране, текст в строке заголовка и др.) установлены по умолчанию (рис. 5.7).

Для окончания исполнения проекта необходимо закрыть окно его исполнения, выбрав кнопку закрытия окна или нажав комбинацию клавиш **Alt + F4**.

В первоначальный проект, как и в любой другой, можно вносить изменения. Например, можно изменить значения некоторых свойств формы. Для этого на вкладке **Свойства** окна **Инспектор объектов** необходимо установить, например:

Caption – *Первое окно*
Color – *clRed* (***color*** – цвет, ***red*** – красный)
Height – *450*
Width – *700*
Left – *600*
Top – *300*

Если сохранить новую версию проекта и снова исполнить его, тоже откроется окно, но другого цвета, других размеров, расположенное в другом месте экрана, с другим текстом в строке заголовка (рис. 5.8).

Чтобы закрыть окно среды **Lazarus**, можно выполнить команду **Выход** из меню **Файл** или выбрать кнопку закрытия окна среды в строке заголовка окна **Lazarus**.

Самое важное в этом пункте

Компьютерная программа – это алгоритм обработки данных, записанный на специальном языке и предназначенный для исполнения компьютером.

В процессе работы программа обрабатывает данные.

Данные, которые передаются в программу с некоторых устройств (например, с клавиатуры или с датчика температуры), из другой программы или другого источника (например, из текстового файла), называют **входными (начальными) данными**. Некоторые программы работают без входных данных.

При исполнении программы создаются и обрабатываются и другие данные, которые называют **промежуточными данными**.

Если целью исполнения программы является получение определенных данных, то эти данные называют **выходными (результирующими) данными**.

Язык, который используют для записи алгоритмов, предназначенных для исполнения компьютером, называют **языком программирования**.

Каждый язык программирования имеет следующие составляющие:

Алфавит – множество символов, из которых можно создавать слова и предложения этого языка.

Словарь – набор слов, которые используются в этом языке.

Синтаксис – правила составления и записи речевых конструкций: не словарных слов и предложений.

Семантику – установленное однозначное толкование языковых конструкций, правил их исполнения.

Среда разработки программ Lazarus состоит из:

- **редактора исходного кода** для ввода и редактирования текста программ;
- **компилятора** для перевода программы с языка программирования **Object Pascal** на язык команд, которые может исполнить процессор компьютера;
- **средств отладки программы** для поиска в ней ошибок;
- **справочной системы**

и других компонентов.

Форма – это объект, на котором можно разместить различные компоненты, например, кнопки, поля, метки, меню, полосы прокрутки и др. В ходе исполнения проекта форма открывается как окно приложения, на котором отображаются все объекты, расположенные на форме.

Примеры некоторых свойств формы:

- **Name** (англ. *name* – имя) – его значение определяет имя формы как объекта;
- **Caption** (англ. *caption* – заголовок) – его значение определяет текст в строке заголовка формы;
- **Color** (англ. *color* – цвет) – его значение определяет цвет фона формы;
- **Height** (англ. *height* – высота) – его значение определяет высоту окна в пикселях (**пиксель** – это точка графического изображения на экране);
- **Width** (англ. *width* – ширина) – его значение определяет ширину окна в пикселях;
- **Top** (англ. *top* – верх) – его значение определяет расстояние (в пикселях) от верхней границы окна до верхней границы экрана;
- **Left** (англ. *left* – левый) – его значение определяет расстояние (в пикселях) от левой границы окна до левой границы экрана.

Дайте ответы на вопросы

1°. Что такое компьютерная программа?

2•. Какие данные программы называют входными, промежуточными, выходными?

3°. Что такое язык программирования?

4•. Какие составляющие части языка программирования существуют? Опишите каждую из них.

5•. Что такое синтаксическая ошибка?

6°. Для чего предназначена среда **Lazarus**?

7•. Какие основные составляющие среды **Lazarus** и каково их назначение?

8•. Для чего предназначена программа-компилятор? Каковы основные этапы ее исполнения?

9°. Что такое форма? Назовите несколько ее свойств.

10°. Как открыть проект в среде **Lazarus**? Как сохранить проект?

Выполните задания

1°. Создайте проект, исполнение которого откроет окно красного цвета размерами 400 × 350 и изменит текст в заголовке окна на ваше имя. Сохраните проект в папке с именем **Задание 5.1.1**, созданной в вашей папке.

2°. Создайте проект, исполнение которого откроет окно синего цвета размерами 200 × 450 и изменит текст в заголовке окна на вашу фамилию. Сохраните проект в папке с именем **Задание 5.1.2**, созданной в вашей папке.

3•. Создайте проект, исполнение которого откроет окно синего цвета размерами 300 × 300, левая граница которого находится на расстоянии 300 пикселей от левой границы экрана и верхняя граница которого находится на расстоянии 400 пикселей от верхней границы экрана. Сохраните проект в папке с именем **Задание 5.1.3**, созданной в вашей папке.

4•. Создайте проект, исполнение которого откроет окно серого цвета размерами 200 × 100, левая граница которого находится на расстоянии 150 пикселей от левой границы экрана и верхняя граница которого находится на расстоянии 100 пикселей от верхней границы экрана. Сохраните проект в папке с именем **Задание 5.1.4**, созданной в вашей папке.

5.2. СОБЫТИЯ. ОБРАБОТЧИКИ СОБЫТИЙ

1. Что такое компьютерная программа?
2. Что такое форма? Назовите несколько ее свойств.
3. Каково предназначение компилятора и каковы этапы его исполнения?

СОБЫТИЯ И ИХ ОБРАБОТЧИКИ

Форма, кроме набора свойств, имеет еще набор **событий**, которые могут с ней произойти при выполнении проекта.

Такими событиями, например, являются:

- **Click** (англ. *click* – щелчок) – выбор любой точки объекта;
- **Close** (англ. *close* – закрывать) – закрытие формы (окна) при завершении исполнения проекта;
- **Create** (англ. *create* – создание) – создание (открытие) формы (окна) в начале исполнения проекта;
- **KeyPress** (англ. *key press* – нажатие клавиши) – нажатие любой клавиши на клавиатуре;
- **MouseMove** (англ. *mouse move* – перемещение мыши) – перемещение указателя по окну

и другие.

С любым событием, которое может произойти с формой, можно связать фрагмент программы, который будет исполняться сразу после наступления этого события. Такой фрагмент программы называют **обработчиком**

события. Обработчик события, связанный с определенным объектом, называют **методом этого объекта**.

Среда **Lazarus** по умолчанию создает обработчик события как **процедуру**. **Процедура** – это отдельно выделенная часть программы, имеющая свое имя и которую по этому имени можно запустить на исполнение.

Исполнение процедуры также является событием. Если процедура исполняется в результате наступления определенного события с формой, например события **Click**, то событие самого исполнения этой процедуры называют **OnClick** (англ. *on* – на).

Например, если необходимо связать некоторую процедуру с событием **Click** для формы, то следует:

1. Выбрать форму.
2. Открыть вкладку **События** в окне **Инспектор объектов**.
3. Выбрать событие **OnClick**. Поле справа от текста **OnClick** станет текущим и в нем появится курсор.
4. Щелкнуть дважды в этом текущем поле.

В результате исполнения этого алгоритма в поле **OnClick** на вкладке **События** появляется текст **FormClick** (англ. *form click* – щелчок формы), а в окне редактора исходного кода текста проекта на вкладке **Unit1** появляется заготовка для текста процедуры с именем **TForm1.FormClick**, команды которой и будут исполняться при наступлении события **Click** с формой (рис. 5.9).

Команды процедуры записываются между двумя словами **begin** (англ. *begin* – начало) и **end** (англ. *end* – конец). *Обращаем ваше внимание*, что **begin** и **end** – это не команды, эти слова играют роль соответственно открывающей и закрывающей скобки, определяющие начало и окончание процедуры. Эти слова часто называют **операторными скобками**.

После слова **end** в конце процедуры должна стоять точка с запятой (**;**). Каждая команда процедуры также должна заканчиваться точкой с запятой (**;**).

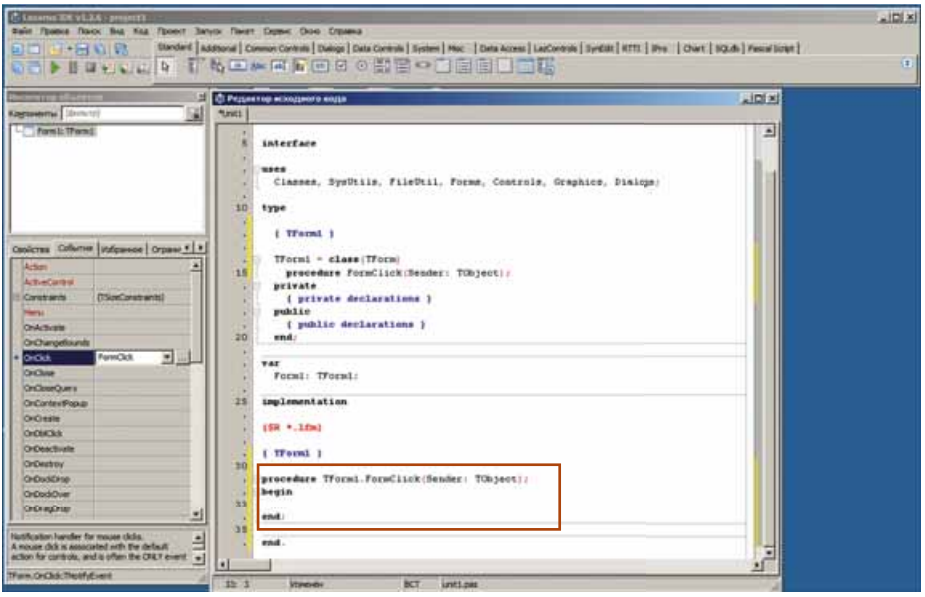


Рис. 5.9. Заготовка для текста процедуры **TForm1.FormClick**

Если необходимо, чтобы в результате исполнения процедуры изменилось значение определенного свойства формы, процедура должна содержать **команду присваивания**. Она может иметь следующий вид:

<имя объекта>.<название свойства>:= <значение или выражение>;

или:

<имя объекта>.<название свойства>:= <имя объекта>.<название свойства> + <значение или выражение>;

Например, исполнение команды **Form1.Color := clGreen** установит зеленым цвет фона формы, а команды **Form1.Top := Form1.Top + 50** – увеличит текущий отступ верхней границы окна от верхней границы экрана на 50 пикселей. При исполнении последней команды сначала к текущему значению свойства **Top** формы будет прибавлено число 50, после чего полученное число станет новым значением этого свойства формы.

Командой присваивания **Form1.Caption := 'Изменение значений свойств формы'** можно установить указанный в кавычках текст в строке заголовка формы, а командой **Form1.Left := 300** – установить отступ 300 пикселей левой границы окна от левой границы экрана.

Приведем пример процедуры **TForm1.FormClick**, состоящей из команд, приведенных выше, в результате исполнения которой изменятся значения свойств **Caption, Color, Left** и **Top** формы, а также результат ее исполнения (рис. 5.10).

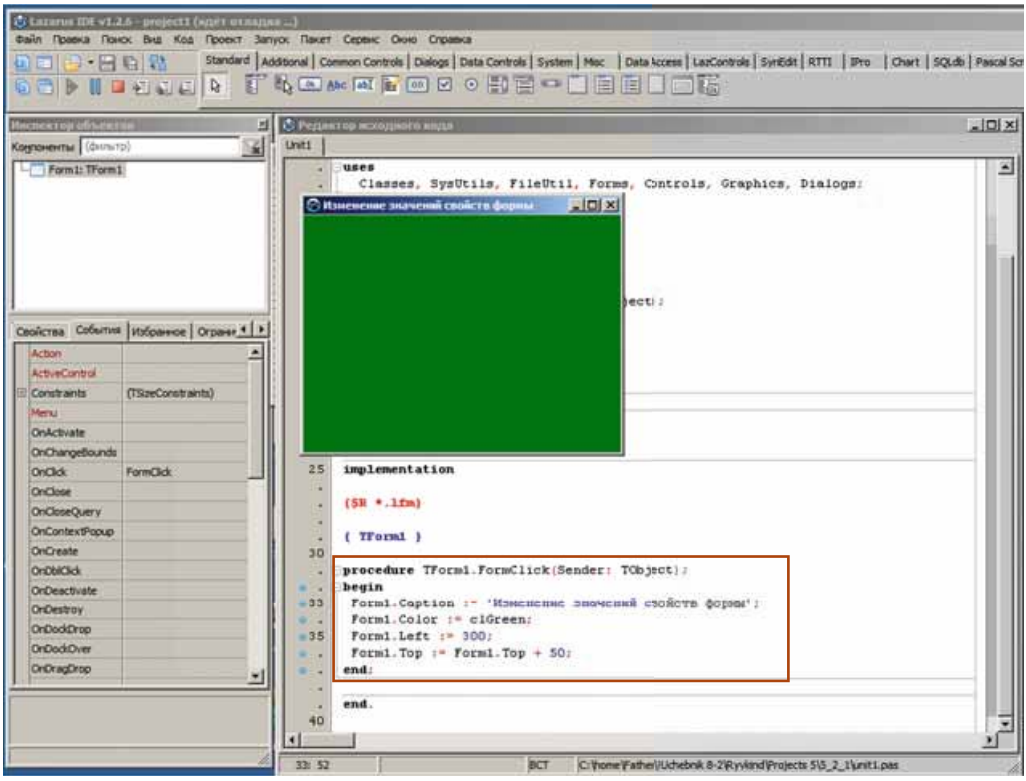


Рис. 5.10. Процедура изменения значения свойств **Caption**, **Color**, **Left** и **Top** формы и результат ее исполнения

После запуска проекта на исполнение необходимо дождаться открытия окна его исполнения и выбрать любую его точку. Таким образом, произойдет событие **Click** для формы и как следствие произойдет событие **OnClick**, которое будет заключаться в исполнении процедуры **TForm1.FormClick** – обработчика события **Click** для формы.

СРЕДСТВА СОЗДАНИЯ И РЕДАКТИРОВАНИЯ ПРОЕКТА

Среда **Lazarus** имеет удобные средства помощи при написании текста проекта:

- после ввода первой или нескольких первых букв имени объекта можно нажать комбинацию клавиш **Ctrl + Пробел** и откроется список, из которого можно выбрать необходимое имя объекта (рис. 5.11);
- после ввода точки, разделяющей имя объекта и название его свойства, автоматически открывается окно со списком, который содержит названия свойств данного объекта (рис. 5.12). Дальше название свойства можно не вводить с клавиатуры, а найти в списке и нажать клавишу **Enter**.

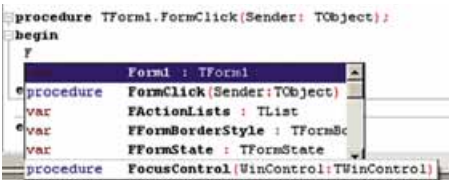


Рис. 5.11. Список для выбора имени объекта

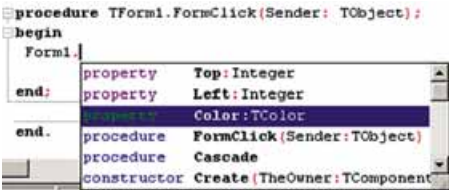


Рис. 5.12. Список с названиями свойств объекта

Можно после открытия списка ввести несколько первых букв из названия свойства. Тогда этот список будет сокращаться, и в нем будут оставаться только те названия, которые начинаются с введенных букв. Этот способ ускоряет ввод названий свойств объектов, а также предотвращает синтаксические ошибки, особенно для тех, кто не учил английский язык.

Во многих случаях система автоматически расставляет в необходимых местах текста программы операторные скобки, знак присваивания, точку с запятой в конце команды и т. п.

Отметим также, что система не различает большие и малые буквы. Однако, если слово образовано из нескольких слов, принято (и удобно для лучшего понимания) вводить первую букву каждого слова большой.

Вы уже знаете, что в начале исполнения проекта программа-компилятор проверяет текст проекта на наличие синтаксических ошибок. Если такие ошибки есть, то исполнение проекта прерывается и в окне **Messages** (англ. *messages* – сообщения), которое открывается под окном **Редактора исходного кода**, появляются сообщения-подсказки о номерах строк и номерах позиций в них, где допущены ошибки, а также комментарии к каждой из этих ошибок. Кроме того, цветом выделяется фон первой из тех строк текста проекта, содержащей ошибку, или цвет фона строки, следующей за строкой с ошибкой.

Так, на рисунке 5.13 выделена цветом строка 34, так как именно в ней содержатся первые синтаксические ошибки проекта – неправильно напи-

саны слова **Color** и **Green**. Кроме того, в конце команды, которая находится в строке 35, пропущена точка с запятой. Для такой ошибки в окне **Messages** появляется сообщение о неправильном символе, с которого начинается следующая, 36-я, строка. И если именно эта ошибка была бы в проекте первой, то цветом была бы выделена именно следующая, 36-я, строка.

Необходимо исправить в тексте проекта все синтаксические ошибки и снова запустить проект на исполнение.

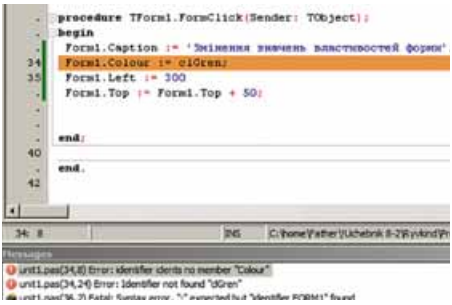


Рис. 5.13. Обозначение синтаксических ошибок в среде **Lazarus**

ОКНО СООБЩЕНИЙ И ЕГО ИСПОЛЬЗОВАНИЕ

Для вывода на экран текстовых сообщений можно использовать специальное окно – **окно сообщений**. Для открытия такого окна и вывода в нем необходимого текстового сообщения используется команда **ShowMessage ('<текст>')** (англ. *show message* – показывать сообщение).

Пример текста процедуры – обработчика события **MouseMove** (англ. *mouse move* – перемещение мыши) с такой командой и результат исполнения соответствующего проекта приведен на рисунке 5.14.

После ознакомления с сообщением следует закрыть это окно, выбрав кнопку **OK** или кнопку закрытия окна. Только после этого можно закрыть окно исполнения проекта.

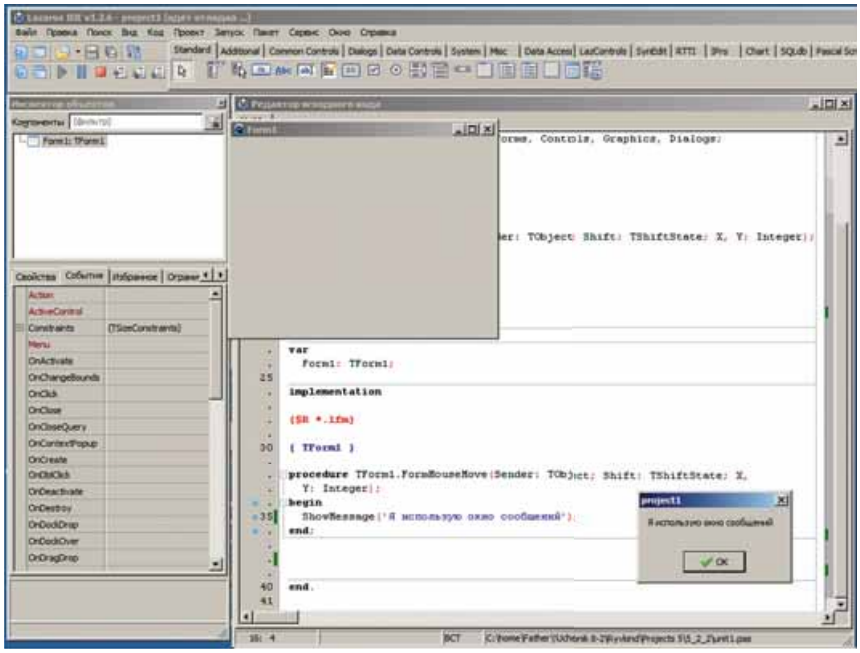


Рис. 5.14. Использование окна сообщений

Обращаем ваше внимание, в этом примере использовано событие **MouseMove**. Интересно, что в этом случае окно исполнения проекта не удастся закрыть, используя мышь. Ведь как только указатель будет перемещаться по окну, наступит событие **MouseMove** и как результат – событие **OnMouseMove**, то есть запустится на исполнение обработчик события **MouseMove** – процедура **TForm1.FormMouseMove**, которая вновь откроет окно сообщений. Поэтому закончить исполнение этого проекта можно, нажав комбинацию клавиш **Alt + F4**. Понятно, что эту комбинацию клавиш можно использовать для закрытия окна исполнения любого проекта.

Самое важное в этом пункте

С любым событием, которое может произойти с формой, можно связать фрагмент программы, который будет исполняться сразу после наступления этого события. Такой фрагмент программы называют **обработчиком события**. Обработчик события, связанный с определенным объектом, называют **методом этого объекта**.

Среда **Lazarus** по умолчанию оформляет обработчик события как процедуру. **Процедура** – это отдельно выделенная часть программы, которая имеет свое имя и которую по этому имени можно запустить на исполнение.

Команды процедуры записываются между двумя словами **begin** и **end**. Эти слова играют роль соответственно открывающей и закрывающей скобки **(операторные скобки)**, определяющие начало и окончание процедуры.

После слова **end** в конце процедуры должна стоять точка с запятой **(;)**. Каждая команда процедуры также должна заканчиваться точкой с запятой **(;)**.

Если необходимо, чтобы в результате исполнения процедуры изменилось значение определенного свойства формы, процедура должна содержать **команду присваивания**. Она может иметь следующий вид:

<имя объекта>.<название свойства> := <значение или выражение>;

или такой:

<имя объекта>.<название свойства> := <имя объекта>.<название свойства> + <значение или выражение>;

Для открытия окна сообщений и вывода в нем необходимого текстового сообщения используется команда **ShowMessage ('<текст>');**.

Дайте ответы на вопросы

1°. Какие события могут происходить с формой?
2•. Какая связь между событиями **Click** и **OnClick**?
3•. Что называют обработчиком события?
4•. Как связать с событием его обработчик?
5•. Что называют методом объекта?
6°. Что называют процедурой?
7°. Что называют операторными скобками? Для чего они используются?
8•. Какими командами можно изменить значения свойств объекта? Каков их общий вид? Приведите примеры таких команд.
9•. Какой командой можно открыть окно сообщений?

Выполните задания

1°. Создайте проект, в котором событие **Click** для формы установит ее размеры 400 × 300 пикселей, зеленый цвет фона и изменит текст заголовка формы на название вашего города. Сохраните проект в папке с именем **Задание 5.2.1**, созданной в вашей папке.

2°. Создайте проект, в котором событие **KeyPress** для формы установит ее размеры 300 × 200 пикселей, синий цвет фона и изменит текст заголовка формы на ваше имя. Сохраните проект в папке с именем **Задание 5.2.2**, созданной в вашей папке.

3•. Создайте проект, в котором событие **DblClick** (англ. ***Double*** – двойной) для формы повлечет изменение цвета ее фона, заголовка и увеличит на 200 пикселей отступ ее верхней границы от верхней границы экрана. Сохраните проект в папке с именем **Задание 5.2.3**, созданной в вашей папке.

4•. Создайте проект, в котором событие **Show** (англ. *show* – показывать) для формы вызовет изменение ее размеров, цвета фона и уменьшит на 100 пикселей отступ ее левой границы от левой границы экрана. Сохраните проект в папке с именем **Задание 5.2.4**, созданной в вашей папке.

5•. Создайте проект, в котором событие **Click** для формы установит ее размеры 500 × 500 пикселей, коричневый цвет фона и изменит текст заголовка формы на «Проект с двумя процедурами», а событие **DblClick** откроет окно сообщений с текстом «Я – гражданин Украины!». Сохраните проект в папке с именем **Задание 5.2.5**, созданной в вашей папке.

6•. Создайте проект, в котором событие **Close** для формы откроет окно сообщений с текстом «Выполнение проекта завершено! До свидания!». Сохраните проект в папке с именем **Задание 5.2.6**, созданной в вашей папке.

ПРАКТИЧЕСКАЯ РАБОТА № 8

«Создание объектно-ориентированной программы, отображающей окно сообщения»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте окно среды **Lazarus**.
2. Установите следующие значения свойств формы:
   - цвет фона – *серый*;
   - ширина – *200 пикселей*;
   - высота – *100 пикселей*;
   - отступ левой границы – *150 пикселей*;
   - отступ верхней границы – *100 пикселей*;
   - текст в строке заголовка – *Практическая № 8*.
3. Сохраните проект в папке с именем **Практическая 8_1**, созданной в вашей папке.
4. Исполните проект.

5. Закройте окно исполнения проекта.
6. Создайте обработчик события **Click** для формы, исполнение которого установит красный цвет фона окна, установит отступ верхней границы окна 200 пикселей от верхней границы экрана, увеличит его ширину на 300 пикселей, уменьшит на 50 пикселей отступ левой границы окна от левой границы экрана, откроет окно сообщений с текстом «Мы изучаем язык программирования **Object Pascal**».
7. Сохраните проект в папке с именем **Практическая 8_2**, созданной в вашей папке.
8. Исполните проект.
9. Закройте окно исполнения проекта.
10. Добавьте к проекту обработчик события **KeyPress** для формы, исполнение которого изменит цвет фона окна на синий, увеличит на 300 пикселей отступ верхней границы окна от верхней границы экрана, увеличит его ширину на 200 пикселей.
11. Сохраните проект в папке с именем **Практическая 8_3**, созданной в вашей папке.
12. Исполните проект.
13. Закройте окно исполнения проекта.
14. Закройте окно среды **Lazarus**.

5.3. КНОПКА

1. Какие свойства формы вы знаете? Как изменить значения свойств формы до начала исполнения проекта и во время его исполнения?
2. Какие события могут происходить с формой (окном)? В чем суть каждого из них?
3. Что такое обработчик события? Как его создать? Как его исполнить?

КНОПКА, ЕЕ СВОЙСТВА И ОБРАБОТЧИКИ СОБЫТИЙ ДЛЯ КНОПКИ

Одним из компонентов, который можно разместить на форме, является **кнопка**.

Чтобы разместить кнопку на форме, необходимо подвести указатель к ее изображению  на вкладке **Standard** (англ. *standard* – стандарт) окна среды **Lazarus** и дважды щелкнуть левую кнопку мыши. После этого кнопка появится в верхней левой части формы (рис. 5.15), а ее имя добавится в список компонентов проекта в поле **Компоненты** окна **Инспектор объектов**. При необходимости кнопку можно перетащить в любое место формы.

Если выбрать кнопку на форме или выбрать ее имя в поле **Компоненты** окна **Инспектор объектов**, то на вкладке **Свойства** этого окна отображается список свойств кнопки и их значений. Как и для формы, значения некоторых свойств кнопки установлены по умолчанию. Значения свойств кнопки можно изменять. Для некоторых свойств их значения необходимо

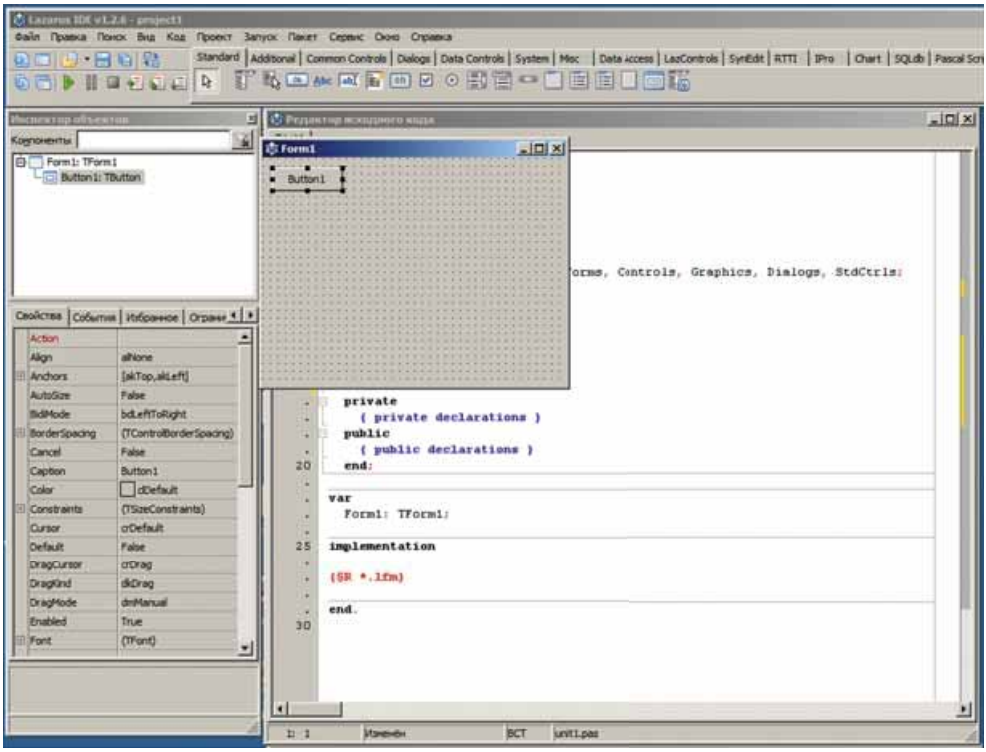


Рис. 5.15. Кнопка на форме

вводить в соответствующие поля, для других – можно выбирать из списка. Значения некоторых свойств можно изменять перетаскиванием кнопки по форме или изменением ее размеров, используя маркеры на ее границах.

Все свойства формы, которые мы рассматривали **(Caption, Color, Height, Width, Top, Left, Name)**, есть и у кнопки. Но изменение значения свойства **Color,** которое установлено по умолчанию, не изменяет стандартный цвет кнопки, значение свойства **Caption** определяет текст на самой кнопке, а значения свойств **Top** и **Left** определяют расстояние верхней и левой границ кнопки не от соответствующих границ экрана, как для формы, а от соответствующих границ формы, на которой расположена эта кнопка.

Рассмотрим еще несколько свойств, которые имеют и кнопка, и форма:

- **Enabled** (англ. *enabled* – доступный) – его значение определяет, доступен ли объект для операций с ним. Это свойство может принимать только одно из двух значений: **True** (англ. *true* – истина) – объект доступен или **False** (англ. *false* – ложь) – объект недоступен.
- **Visible** (англ. *visible* – видимый) – его значение определяет, видимый ли данный объект в окне. Это свойство может также принимать только одно из двух значений: **True** – объект видимый или **False** – объект невидимый.
- **Font** (англ. *font* – шрифт) – его значение определяет значения свойств шрифта, которым будет выводиться текст на объекте. В отличие от всех предыдущих свойств, это свойство является **комплексным**. Так его на-

зывают потому, что оно состоит из нескольких свойств шрифта: **Color**, **Height**, **Name**, **Size** (англ. *size* – размер) и др. Значение каждого из них можно изменить, если открыть их список выбором кнопки ⊞ слева от названия свойства **Font**, или в окне **Шрифт** (рис. 5.16), которое открывается выбором кнопки ... в правой крайней части строки **Font**. Эта кнопка становится доступной, если сделать текущим поле этой строки.

Изменить, например, цвет шрифта в ходе исполнения проекта можно командой **Button1.Text.Color := clBlue**.

Если выделить кнопку, то на вкладке **События** окна **Инспектор объектов** можно, как и для формы, выбрать событие, для которого будет создана заготовка для текста соответствующей процедуры – обработчика этого события. Как и для формы, можно ввести в эту процедуру команды изменения значений свойств кнопки и формы и исполнить созданный проект.

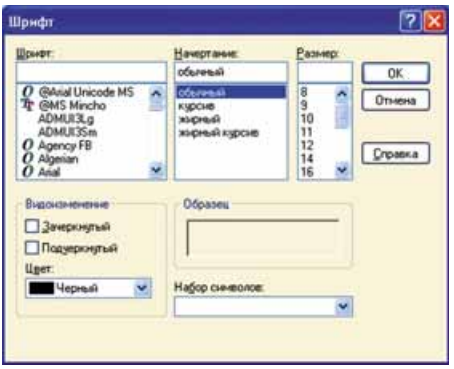


Рис. 5.16. Окно **Шрифт** для установки значений свойств шрифта

На рисунке 5.17 приведен пример обработчика события **Click** для кнопки, содержащий команды:

- задания отступа 150 пикселей левой границы кнопки от левой границы окна;
- увеличения на 50 пикселей отступа верхней границы кнопки от верхней границы окна;

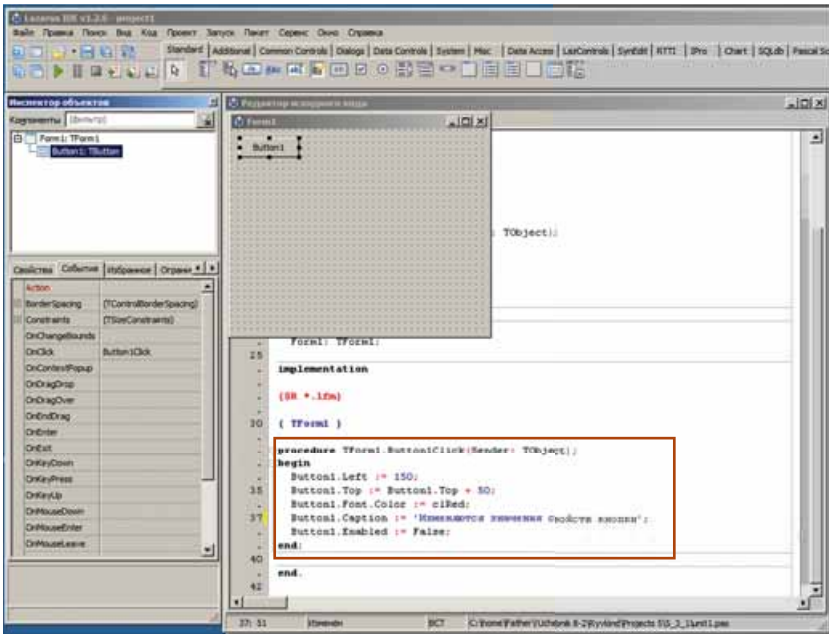


Рис. 5.17. Обработчик события **Click** для кнопки

- задания красного цвета для шрифта, которым будет выводиться текст на кнопке;
- вывода на кнопке указанного текста;
- задания для кнопки режима «недоступна».

Самое важное в этом пункте

Чтобы разместить кнопку на форме, необходимо подвести указатель к ее изображению  на вкладке **Standard** окна среды **Lazarus** и дважды щелкнуть левую кнопку мыши.

Кнопка, как и форма, имеет свойства **Caption, Color, Height, Width, Top, Left, Name.** Но изменение значения свойства **Color**, которое установлено по умолчанию, не изменяет стандартный цвет кнопки, значение свойства **Caption** определяет текст на самой кнопке, а значения свойств **Top** и **Left** определяют расстояние верхней и левой границ кнопки не от соответствующих границ экрана, как для формы, а от соответствующих границ формы, на которой расположена эта кнопка.

Кнопка, как и форма, имеет также свойства:

- **Enabled** – его значение определяет, доступен ли объект для операций с ним. Это свойство может принимать только одно из двух значений: **True** (англ. *true* – истина) – объект доступен или **False** (англ. *false* – недостаток, ложь) – объект недоступен;
- **Visible** – его значение определяет, видимый ли данный объект в окне. Это свойство может также принимать только одно из двух значений: **True** – объект видимый или **False** – объект невидимый.
- **Font** – его значение определяет значение свойств шрифта, которым будет выводиться текст на объекте. Это свойство является **комплексным.** Так его называют потому, что оно состоит из нескольких свойств шрифта: **Color, Height, Name, Size** и др.

Дайте ответы на вопросы

1°. Как разместить кнопку на форме?
2°. Какие свойства кнопки вы знаете?
3•. Что определяют значения известных вам свойств кнопки?
4•. Чем отличается изменение значений свойства **Color** для формы и для кнопки?
5•. Чем отличается свойство **Caption** для формы и для кнопки?
6•. Чем отличаются свойства **Top** и **Left** для формы и для кнопки?
7•. Какие значения могут принимать свойства кнопки **Enabled** и **Visible**? Что определяет каждое из этих значений для каждого из этих свойств?
8•. Почему свойство **Font** называют комплексным?

Выполните задания

1°. Создайте проект, в котором событие **Click** для кнопки установит ее размеры 40 × 30 пикселей и изменит текст на кнопке на название вашего города. Сохраните проект в папке с именем **Задание 5.3.1**, созданной в вашей папке.

2°. Создайте проект, в котором событие **KeyPress** для кнопки установит ее размеры 20 × 40 пикселей и изменит текст на кнопке на ваше имя. Сохраните проект в папке с именем **Задание 5.3.2**, созданной в вашей папке.

3•. Создайте проект, в котором событие **MouseMove** для кнопки изменит текст на кнопке и увеличит отступ ее верхней границы на 20 пикселей. Сохраните проект в папке с именем **Задание 5.3.3**, созданной в вашей папке.

4•. Создайте проект, в котором событие **Click** для кнопки изменит ее размеры и уменьшит отступ ее левой границы на 20 пикселей. Сохраните проект в папке с именем **Задание 5.3.4**, созданной в вашей папке.

5•. Создайте проект, в котором событие **Click** для кнопки установит ее размеры 50 × 50 пикселей, установит коричневый цвет текста на кнопке и изменит текст на кнопке на «Мой проект», а событие **KeyPress** переместит кнопку на 30 пикселей вправо и на 40 пикселей вниз. Сохраните проект в папке с именем **Задание 5.3.5**, созданной в вашей папке.

6*. Создайте проект, в котором расположите на форме три кнопки. Установите такие значения их свойств, чтобы в начале исполнения проекта первая кнопка была видимая и доступная, вторая – видимая и недоступная, третья – невидимая и недоступная. Для каждой кнопки создайте обработчик события **Click**, чтобы:

- выбор первой кнопки делал ее невидимой и недоступной, вторую кнопку – видимой и доступной, третью кнопку – видимой и недоступной;
- выбор второй кнопки делал ее невидимой и недоступной, третью кнопку – видимой и доступной, первую кнопку – видимой и недоступной;
- выбор третьей кнопки делал ее невидимой и недоступной, первую кнопку – видимой и доступной, вторую кнопку – видимой и недоступной.

Сохраните проект в папке с именем **Задание 5.3.6**, созданной в вашей папке.

5.4. МЕТКА

1. Какие свойства кнопки вы знаете? Как изменить значения этих свойств до начала исполнения проекта и во время его исполнения?
2. Какие события могут произойти с кнопкой? В чем заключается каждое из них?
3. Какие различия свойств кнопки и формы вы знаете?

МЕТКА, ЕЕ СВОЙСТВА И ОБРАБОТЧИКИ СОБЫТИЙ ДЛЯ МЕТКИ

Еще одним компонентом, который можно разместить на форме, является **метка**. Этот компонент используется для вывода текстовых сообщений.

Чтобы разместить метку на форме, следует подвести указатель к кнопке **TLabel** (англ. *label* – метка) на вкладке **Standard** окна среды **Lazarus** и дважды щелкнуть левую кнопку мыши. После этого метка появится в верхней левой части формы (рис. 5.18), а ее имя добавится в список компонентов проекта в поле **Компоненты** окна **Инспектор объектов**. При необходимости метку можно перетащить в любое место формы.

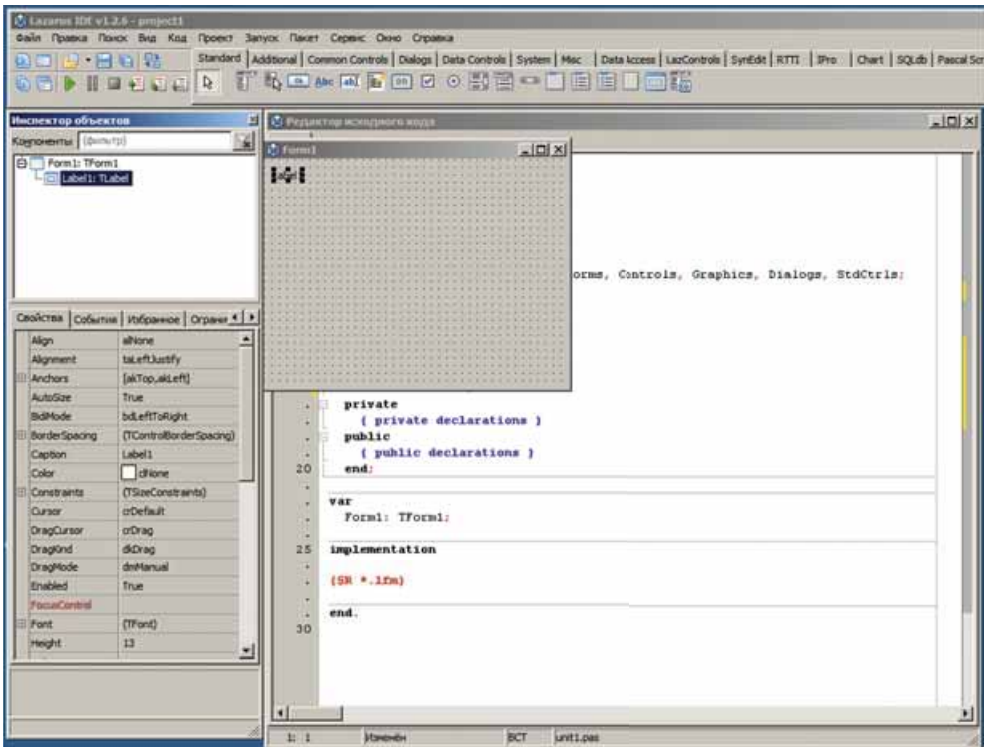


Рис. 5.18. Метка на форме

Выделить метку можно, выбрав ее на форме или выбрав ее имя в поле **Компоненты** окна **Инспектор объектов**. При выбранной метке на вкладке **Свойства** этого окна отображается список ее свойств и их значений. Значения свойств метки можно изменять. Для некоторых свойств их значения необходимо вводить в соответствующие поля, для других – можно выбирать из списка. Значения некоторых свойств можно также изменять перетаскиванием метки по форме.

Свойства метки **Top, Left, Name, Enabled, Visible, Font** аналогичны соответствующим свойствам кнопки. Значение свойства **Color** метки определяет цвет ее фона. Изменение значения этого свойства приводит к изменению цвета фона метки, как это происходит для формы.

В отличие от формы и кнопки, значения свойств **Height** и **Width** для метки нельзя изменить на вкладке **Свойства** или используя указатель. Высота и ширина метки определяются текстом, находящимся в ней. Если изменить текст в поле **Caption**, то в зависимости от его длины изменится ширина метки.

Текст метки можно вводить и редактировать как в поле **Caption**, так и в окне **Диалог ввода строк** (рис. 5.19). В этом окне можно задавать также и многострочные тексты в метке.

Если выделить метку, то на вкладке **События** окна **Инспектор объектов** можно, как и для формы и кнопки, выбрать событие, для которого система создаст заготовку процедуры – обработчика этого события (рис. 5.20). Как и для формы и кнопки, можно ввести в эту процедуру команды изменения значений свойств метки, а также исполнить созданный проект.

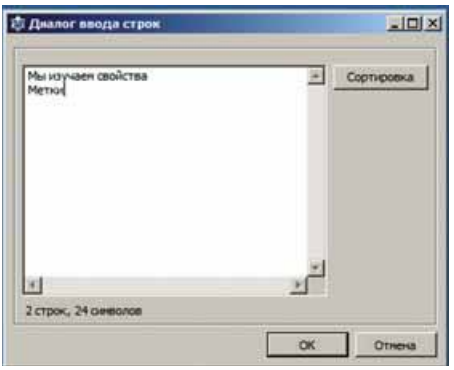


Рис. 5.19. Задание текста метки в окне **Диалог ввода строк**

Приведенная на рисунке 5.20 процедура – обработчик события **Click** для метки – содержит команды:

- увеличения на 20 пикселей отступа верхней границы метки от верхней границы окна;
- задания серого фона метки;
- задания размера 14 шрифта для вывода текста метки;
- задания красного цвета шрифта для вывода текста метки;
- вывода в метку указанного текста.

Результат исполнения этой процедуры приведен на рисунке 5.21.

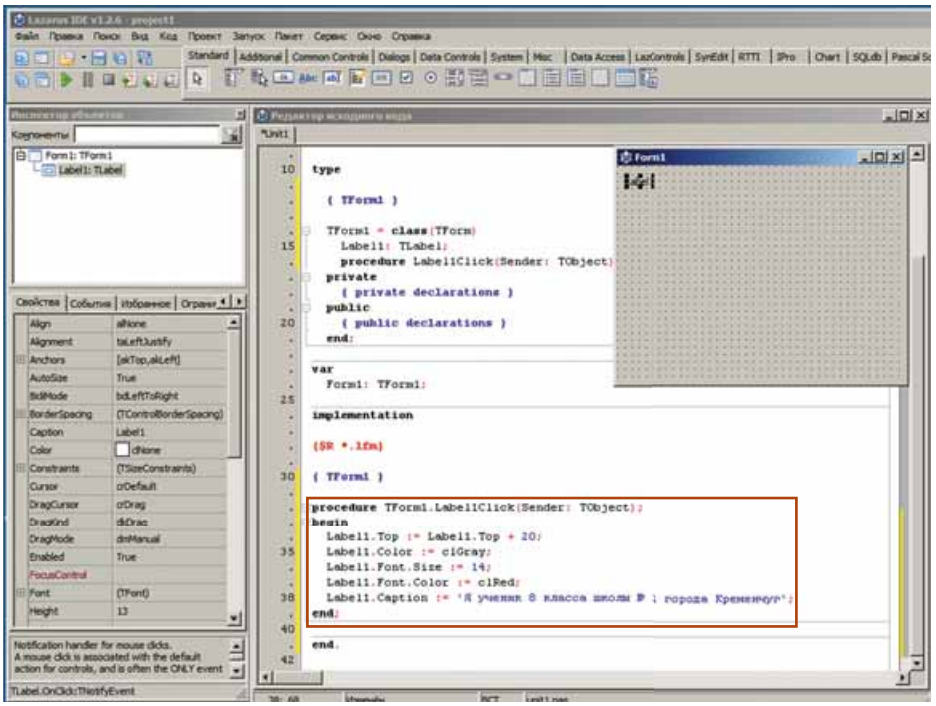


Рис. 5.20. Обработчик события **Click** для метки



Рис. 5.21. Результат исполнения процедуры, приведенной на рисунке 5.20

Для тех, кто хочет знать больше

Чтобы в команде вывода текста в метку текст в необходимом месте переносился на новую строку, следует вставить в команду символ перехода на новую строку (#13).

Например, если в приведенном на рисунке 5.20 примере команду вывода текста в метку изменить на следующую:

Label1.Caption := 'Я ученик 8 класса школы № 1' + #13 + 'города Кременчуг';, то результат исполнения этой процедуры будет выглядеть, как на рисунке 5.22.

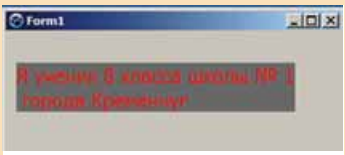


Рис. 5.22. Результат исполнения процедуры, приведенной на рисунке 5.20, с переносом текста

Самое важное в этом пункте

Метка – это компонент, который используется для вывода текстовых сообщений.

Чтобы разместить метку на форме, следует подвести указатель к кнопке **TLabel** на вкладке **Standard** окна среды **Lazarus** и дважды щелкнуть левую кнопку мыши.

Свойства метки **Top, Left, Name, Enabled, Visible, Font** аналогичны соответствующим свойствам кнопки. Значение свойства **Color** метки определяет цвет ее фона. Изменение значения этого свойства приводит к изменению цвета фона метки, как это происходит для формы.

В отличие от формы и кнопки, значения свойств **Height** и **Width** для метки нельзя изменить на вкладке **Свойства** или используя указатель. Высота и ширина метки определяются текстом, расположенным в этой метке. Если изменить текст в поле **Caption**, то в зависимости от длины изменится ширина метки.

Текст метки можно вводить и редактировать как в поле **Caption**, так и в окне **Диалог ввода строк**.

Выделить метку можно, выбрав ее на форме или выбрав ее имя в поле **Компоненты** окна **Инспектор объектов**.

Если выделить метку, то на вкладке **События** окна **Инспектор объектов** можно выбрать событие, для которого система создаст заготовку процедуры – обработчика этого события.

Дайте ответы на вопросы

1°. Для чего используется метка?
2°. Как разместить метку на форме?
3•. Что определяют значения свойств **Top, Left, Name** для метки?
4•. Что определяют значения свойств **Enabled** и **Visible** для метки?
5•. Что определяют значения комплексного свойства **Font** для метки?
6•. Чем отличается свойство **Color** для метки и для кнопки?
7•. Чем отличаются свойства **Top** и **Left** для метки и для формы?
8•. Чем отличаются свойства **Height** и **Width** для метки и для формы?
9•. Как изменить текст в метке?

Выполните задания

1°. Создайте проект, в котором событие **Click** для метки установит 120 пикселей отступ ее верхней границы от верхней границы окна и изменит текст в метке на название улицы, на которой находится ваша школа. Сохраните проект в папке с именем **Задание 5.4.1**, созданной в вашей папке.

2°. Создайте проект, в котором событие **KeyPress** для метки установит 80 пикселей отступ ее левой границы от левой границы окна и изменит текст в метке на ваше имя. Сохраните проект в папке с именем **Задание 5.4.2**, созданной в вашей папке.

3•. Создайте проект, в котором событие **MouseMove** для метки изменит текст в ней, установит красный цвет текста и увеличит отступ метки от верхней границы окна на 20 пикселей. Сохраните проект в папке с именем **Задание 5.4.3**, созданной в вашей папке.

4•. Создайте проект, в котором событие **Click** для метки уменьшит ее отступ от левой границы окна на 20 пикселей, установит размер шрифта 12, цвет шрифта синий и выведет в метку название вашей школы. Сохраните проект в папке с именем **Задание 5.4.4**, созданной в вашей папке.

5•. Создайте проект, в котором событие **Click** для метки установит коричневый цвет фона и изменит текст в метке на «Мой проект», а событие **KeyPress** для метки переместит ее на 30 пикселей вправо и на 40 пикселей вниз. Сохраните проект в папке с именем **Задание 5.4.5**, созданной в вашей папке.

6•. Создайте проект, в котором событие **Click** для кнопки переместит ее на 60 пикселей вправо и на 10 пикселей вверх и выведет в метку вашу фамилию и имя зеленым цветом на сером фоне. Сохраните проект в папке с именем **Задание 5.4.6**, созданной в вашей папке.

7•. Создайте проект, в котором событие **Click** для кнопки переместит окно на 60 пикселей вправо и на 80 пикселей вверх и выведет в метку вашу фамилию и имя серым цветом на зеленом фоне. Сохраните проект в папке с именем **Задание 5.4.7**, созданной в вашей папке.

ПРАКТИЧЕСКАЯ РАБОТА № 9

«Создание программы с кнопками и метками»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте окно среды **Lazarus.**
2. Разместите на форме две кнопки и метку.
3. Установите следующие значения свойств первой кнопки:
   - ширина – *60 пикселей*;
   - высота – *20 пикселей*;
   - отступ от левой границы формы – *120 пикселей*;
   - отступ от верхней границы формы – *100 пикселей*;
   - текст на кнопке – *Форма*.
4. Установите следующие значения свойств второй кнопки:
   - ширина – *100 пикселей*;
   - высота – *30 пикселей*;
   - отступ от левой границы формы – *300 пикселей*;
   - отступ от верхней границы формы – *100 пикселей*;
   - текст на кнопке – *Метка*.
5. Установите следующие значения свойств метки:
   - ширина – *120 пикселей*;
   - высота – *40 пикселей*;
   - отступ от левой границы формы – *150 пикселей*;
   - отступ от верхней границы формы – *200 пикселей*;
   - текст в метке – *название вашего класса*.
6. Сохраните проект в папке с именем **Практическая 9_1**, созданной в вашей папке.
7. Исполните проект.
8. Закройте окно исполнения проекта.
9. Создайте обработчик события **Click** для первой кнопки, исполнение которого установит ширину формы 800 пикселей, высоту формы 400 пикселей, цвет фона формы зеленый, ширину этой кнопки 200 пикселей, увеличит ее высоту на 10 пикселей, переместит ее на 50 пикселей влево и на 30 пикселей вверх, сделает ее недоступной.
10. Создайте обработчик события **MouseMove** для второй кнопки, выполнение которого установит красный цвет фона метки, отступ метки от верхней границы окна 200 пикселей, уменьшит отступ метки от левой границы окна на 50 пикселей, установит цвет текста красный и выведет в метку текст «Мы изучаем язык программирования Object Pascal!».
11. Сохраните новую версию проекта.
12. Исполните новую версию проекта.
13. Закройте окно исполнения проекта.
14. Закройте окно среды **Lazarus.**

Глава 6. Алгоритмы работы с объектами и величинами

В этой главе вы узнаете:

- о переменных величинах и константах, их свойствах
- о настройке проектов
- о компонентах **поле, флажок, поле с раскрывающимся списком,** их свойствах и использовании
- об использовании в проектах ветвления, циклов со счетчиком и с предусловием
- о переменных и их свойствах
- о создании графических объектов в ходе исполнения проектов

6.1. ПОЛЕ. ВЕЛИЧИНЫ, ПОСТОЯННЫЕ И ПЕРЕМЕННЫЕ ВЕЛИЧИНЫ. СВОЙСТВА ПЕРЕМЕННЫХ

1. Для чего используется команда присваивания? Какой вид может иметь эта команда?
2. Какие свойства имеет компонент **метка**? Что определяют значения этих свойств?
3. Какие свойства имеет компонент **кнопка**? Что определяют значения этих свойств?

ПОЛЕ, ЕГО СВОЙСТВА И ОБРАБОТЧИКИ СОБЫТИЙ ДЛЯ ПОЛЯ

Еще одним компонентом, который можно разместить на форме, является **поле**. Поле – это компонент, который используется для ввода данных.

Чтобы разместить поле на форме, необходимо подвести указатель к кнопке **TEdit** (англ. *edit* – редактировать) на вкладке **Standard** окна среды **Lazarus** и дважды щелкнуть левую кнопку мыши. После этого поле появится в верхней левой части формы (рис. 6.1), а его имя добавится в список компонентов проекта в поле **Компоненты** окна **Инспектор объектов**. При необходимости поле можно перетащить в любое место формы.

Выделить поле можно, выбрав его на форме или выбрав его имя в поле **Компоненты** окна **Инспектор объектов**. При выбранном поле список его свойств и их значений отображается на вкладке **Свойства** этого окна. Значения свойств поля можно изменять. Для некоторых свойств их значения необходимо вводить в соответствующие поля на вкладке **Свойства**, для

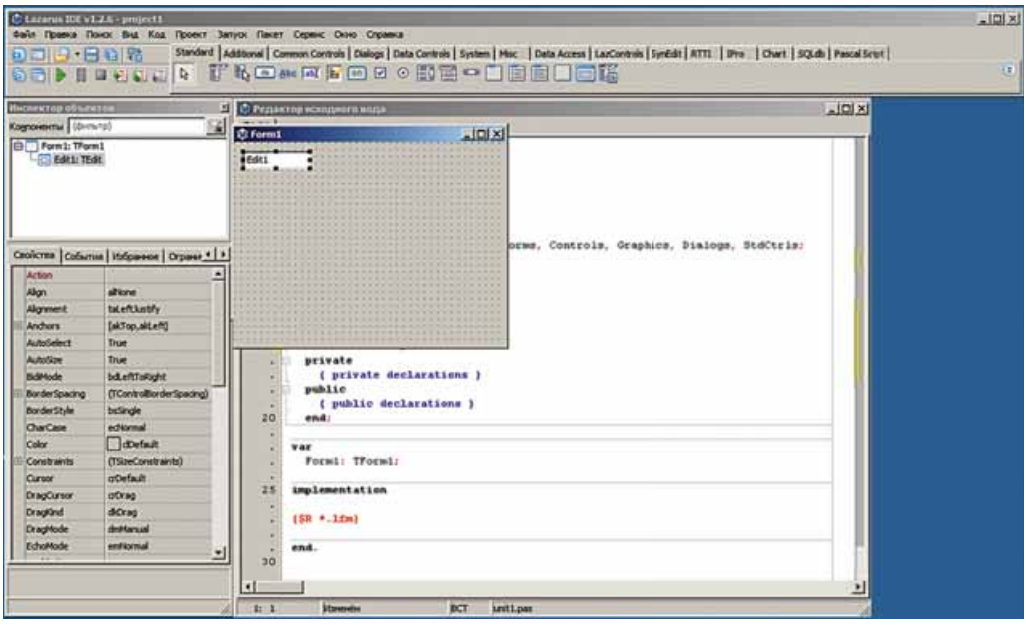


Рис. 6.1. Поле на форме

некоторых – можно выбирать из списка. Их можно также изменять перетаскиванием поля по форме или маркеров на его границах.

Свойства поля **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font** аналогичны соответствующим свойствам метки. Как и для формы, и для кнопки, значения свойств поля **Height** и **Width** можно изменить или на вкладке **Свойства**, или перетаскиванием специальных маркеров на границах поля. Текст в поле является значением его свойства **Text** (англ. *text* – текст). Его можно вводить и редактировать в соответствующем поле вкладки **Свойства**. Часто перед началом исполнения проекта значение этого свойства целесообразно сделать пустым.

Если выделить поле, то на вкладке **События** окна **Инспектор объектов** можно, как и для рассмотренных ранее компонентов, выбрать событие, для которого система создаст заготовку процедуры – обработчика этого события. Приведенная на рисунке 6.2 процедура – обработчик события **Click** для поля – содержит команды:

- уменьшения на 20 пикселей отступа верхней границы поля от верхней границы окна;
- задания желтого фона поля;
- задания размера 12 шрифта для текста в поле;
- задания красного цвета шрифта для текста в поле;
- вывода текста, введенного в поле, в надпись и в окно сообщений.

После запуска проекта на исполнение можно ввести в поле произвольный текст (на рис. 6.3 в поле введен текст «Работаем с полем»), установить указатель на поле и щелкнуть левую кнопку мыши. Результат исполнения процедуры – обработчика события **Click** для поля – приведен на рисунке 6.4.

Несмотря на то что основное назначение поля – ввод данных, оно может использоваться и для вывода данных. Для этого в ходе исполнения проекта значению его свойства **Text** следует присвоить определенный текст. Напри-

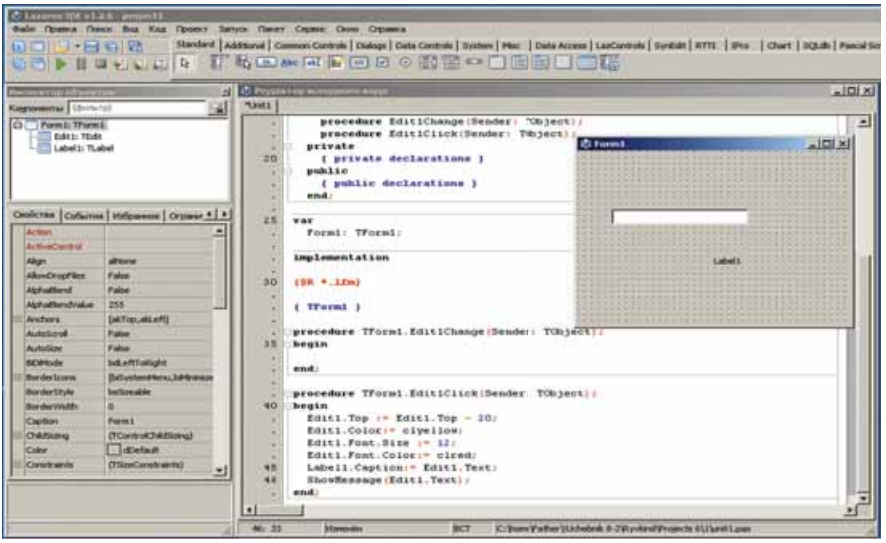


Рис. 6.2. Обработчик события **Click** для поля



Рис. 6.3. Окно исполнения проекта после ввода текста в поле и до наступления события **Click** для поля

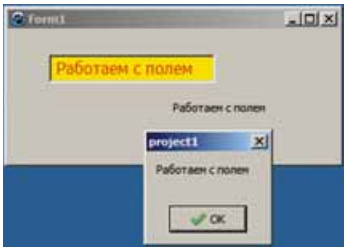


Рис. 6.4. Окно исполнения проекта после наступления события **Click** для поля

мер, **Edit1.Text := 'Мы – ученики 8 класса'.** Если ширина поля будет меньше длины текста, то, в отличие от метки, ширина поля не увеличится и часть текста будет находиться за пределами поля и будет невидима пользователю.

ВЕЛИЧИНЫ. ПОСТОЯННЫЕ И ПЕРЕМЕННЫЕ ВЕЛИЧИНЫ

Информатика, математика, физика и другие науки используют **величины.**

Величина – это количественно выраженное значение свойства объекта.

Примерами величин в информатике являются, например, ширина и высота окна, цвет его фона, надпись на кнопке, ее расположение на форме и др. Примерами величин в математике являются длина и ширина прямоугольника, его площадь, мера угла и др. Примерами величин в физике являются расстояние, время, скорость, плотность вещества и др.

Величину, значение которой не изменяется, называют **постоянной величиной**, или **константой.**

Примером математической константы является, например, число π. Вы знаете, что приближенное значение этого числа равно 3,1416. Примерами физических констант являются плотности веществ, например плотность воздуха приближенно равна 1,293 кг/м^3.

Величину, значение которой может изменяться, называют **переменной величиной**, или **переменной.**

ИСПОЛЬЗОВАНИЕ ПОЛЯ ДЛЯ ВВОДА ЧИСЕЛ

Создадим проект, в котором будет вычисляться сумма двух произвольных действительных чисел.

Разместим на форме два поля, в которые пользователь при исполнении проекта будет вводить слагаемые, метку для вывода суммы и кнопку, после выбора которой будет происходить сложение введенных в поля чисел и вывод суммы в метку.

Форма для такого проекта с размещенными на ней вышеуказанными объектами, а также процедура для сложения двух действительных чисел приведены на рисунке 6.5.

Рассмотрим приведенную процедуру.

Хотя пользователь во время исполнения проекта вводит в поля числа, система воспринимает их как тексты. Эти тексты являются значениями свойства **Text** полей с именами **Edit1** и **Edit2.** Поэтому первые две команды

процедуры **x := StrToFloat (Edit1.Text)** и **y := StrToFloat (Edit2.Text)** (англ. ***string to float*** – строка в число с плавающей десятичной запятой) переводят текстовое представление действительных чисел непосредственно в действительные числа и присваивают их переменным ***x*** и ***y*** соответственно.

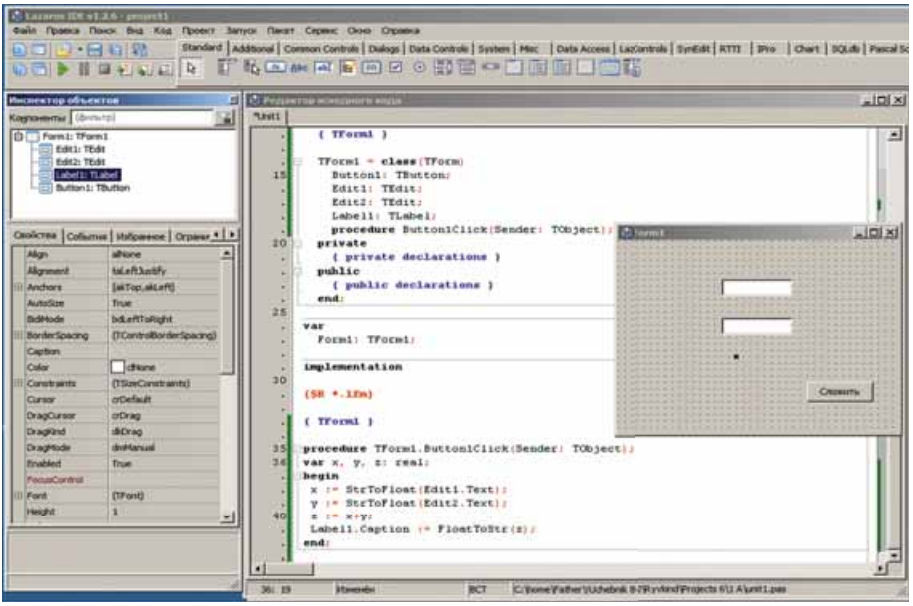


Рис. 6.5. Форма с двумя полями, меткой и кнопкой и процедура для сложения двух действительных чисел

Третья команда складывает эти числа и присваивает сумму переменной ***z***.

В четвертой команде **Label1.Caption := FloatToStr (z)** (англ. ***float to string*** – число с плавающей десятичной запятой в строку) переводит вычисленную сумму из числа в его текстовое представление и присваивает этот текст значению свойства **Caption** метки с именем **Label1**.

В языке программирования **Object Pascal** для любой переменной должен быть указан ее **тип**.

Тип переменной определяет:

- какие значения может принимать эта переменная;
- какие операции с ней можно выполнять;
- какой объем оперативной памяти будет выделен для хранения значения этой переменной.

Имена переменных и их типы указываются в проекте после ключевого слова **var** (англ. *variable* – переменная).

Значения переменных ***x***, ***y***, ***z***, которые использованы в процедуре, по условию задачи могут быть произвольными действительными числами. Для таких переменных можно использовать тип **real** (англ. *real* – действительный). Переменные этого типа могут принимать целые и дробные значения, модуль которых не превышает $1{,}7 \cdot 10^{38}$. Для каждого из этих значений будет выделено 8 байт памяти. Переменные, используемые в процедуре, и их типы описываются в строке **var**, расположенной после строки заголовка процедуры (рис. 6.5).

Если переменные могут принимать только целые значения, то для них можно использовать тип **integer** (англ. *integer* – целый) или тип **longint** (англ. ***long integer*** – длинное целое). Переменные первого из этих типов могут принимать целые значения от –32 768 до 32 767, а второго – целые значения от –2 147 483 648 до 2 147 483 647. Для каждого из этих значений будет выделено 4 байта памяти.



Рис. 6.6. Окно исполнения проекта сложения двух действительных чисел

В строке **var**, расположенной вне процедуры (рис. 6.5), описываются те переменные и их типы, которые могут использоваться во всех процедурах проекта. Если в строке **var** описываются переменные разных типов, то она будет иметь вид, например, такой: **var x, y: real; t: integer;**.

При исполнении созданного проекта открывается окно. Пользователь вводит в поля два числа (на рис. 6.6 это числа 2,7 и –3,43), выбирает кнопку с заголовком **Сложить** и получает результат (–0,73). Не закрывая окна исполнения проекта, пользователь может ввести в поля другие числа, снова выбрать кнопку с заголовком **Сложить** и получить новый результат. И так можно повторять любое количество раз.

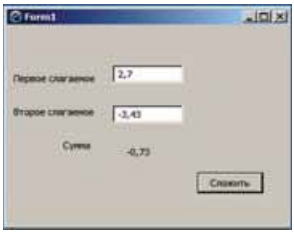


Рис. 6.7. Окно исполнения проекта сложения двух действительных чисел с пояснительными текстами

Чтобы при исполнении проекта пользователю было понятно, что делать, целесообразно на форме слева от каждого поля и от метки для вывода разместить по метке с пояснительными текстами, например: *Первое слагаемое*, *Второе слагаемое*, *Сумма* (рис. 6.7).

ТИП РЕЗУЛЬТАТОВ ОПЕРАЦИЙ НАД ПЕРЕМЕННЫМИ ЧИСЛОВЫХ ТИПОВ

Вы уже знаете, что над переменными числовых типов можно выполнять операции сложения, вычитания, умножения и деления.

Если в командах $x := a + b$, $x := a - b$, $x := a*b$ и $x := a/b$ переменные a и b являются переменными типа **real**, то и тип переменной x должен быть определен как **real**.

Если в командах $x := a + b$, $x := a - b$, $x := a*b$ переменные a и b являются переменными типа **integer (longint)**, то тип переменной x может быть определен и как **integer (longint)**, и как **real**.

А вот если в команде $x := a/b$ переменные a и b являются переменными типа **integer (longint)**, то тип переменной x должен быть определен как **real**.

ИСПОЛЬЗОВАНИЕ КОНСТАНТ В ПРОЕКТЕ

В проекте, кроме переменных, можно использовать константы.

Создадим проект, в котором нужно вычислить массу сосны, если известен ее объем.

Из курса физики известно, что плотность сосны является константой и равна 520 кг/м^3, а массу можно вычислить по формуле $m = \rho V$.

Будем считать, что значение объема вводится в кубических метрах в поле с именем **Edit1**, а значение массы выводится в килограммах в метку с именем **Label1**.

Тогда соответствующий фрагмент проекта будет выглядеть так:

```
const p = 520;
var x, m: real;
begin
  x := StrToFloat(Edit1.Text);
  m := x*p;
  Label1.Caption := FloatToStr(m);
end;
```

Самое важное в этом пункте

Поле – это компонент, который используется для ввода данных.

Свойства поля **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font** аналогичны соответствующим свойствам метки. Значения свойств поля **Height** и **Width** можно изменить или на вкладке **Свойства**, или перетаскиванием маркеров на границах поля. Текст в поле является значением его свойства **Text**. Его можно вводить и редактировать в соответствующем поле вкладки **Свойства**.

Поле может использоваться и для вывода данных. Для этого значению его свойства **Text** необходимо присвоить определенный текст.

Величина – это количественно выраженное значение свойства объекта.

Постоянная величина (константа) – величина, значение которой не изменяется. **Переменная величина (переменная)** – величина, значение которой может изменяться.

В языке программирования **Object Pascal** для любой переменной должен быть указан ее **тип**. **Тип переменной** определяет:

- какие значения может принимать эта переменная;
- какие операции с ней можно выполнять;
- какой объем оперативной памяти будет выделен для хранения значения этой переменной.

В языке программирования **Object Pascal** могут быть использованы константы.

Дайте ответы на вопросы

1°. Для чего может использоваться поле?
2°. Как разместить поле на форме?
3•. Что определяют значения свойств **Top**, **Left**, **Name**, **Color**, **Text**, **Enabled** и **Visible** для поля?
4•. Что определяет значение комплексного свойства **Font** для поля?
5•. Чем отличаются свойства **Top** и **Left** для поля и для формы?
6•. Чем отличаются свойства **Height** и **Width** для поля и для метки?
7•. Как можно изменить текст в поле?
8•. Какими командами можно перевести текстовое представление числа непосредственно в число и обратно?
9•. Что определяет тип переменной?
10•. Где и как указываются имена переменных и их типы?

Выполните задания

1°. Создайте проект, в котором событие **Click** для поля установит 100 пикселей отступ его верхней границы от верхней границы окна и изменит текст в поле на слово «Информатика». Сохраните проект в папке с именем **Задание 6.1.1**, созданной в вашей папке.

2°. Создайте проект, в котором событие **KeyPress** для поля установит 120 пикселей отступ его левой границы от левой границы окна и изменит текст в поле на ваше имя. Сохраните проект в папке с именем **Задание 6.1.2**, созданной в вашей папке.

3•. Создайте проект, в котором событие **MouseMove** для поля изменит текст в нем, установит красный цвет текста, серый цвет фона и переместит поле на 100 пикселей вправо и на 80 пикселей вниз. Сохраните проект в папке с именем **Задание 6.1.3**, созданной в вашей папке.

4•. Создайте проект, в котором событие **Click** для поля переместит его на 80 пикселей вправо и на 20 пикселей вверх и выведет в поле вашу фамилию и имя зеленым цветом на желтом фоне. Сохраните проект в папке с именем **Задание 6.1.4**, созданной в вашей папке.

5•. Создайте проект для вычисления значения выражения $(2a + b) \cdot (a + c)$. Сохраните проект в папке с именем **Задание 6.1.5**, созданной в вашей папке.

6•. Создайте проект для вычитания двух действительных чисел. Сохраните проект в папке с именем **Задание 6.1.6**, созданной в вашей папке.

7•. Создайте проект для вычисления значения выражения $(3a - 2b) \cdot c$. Сохраните проект в папке с именем **Задание 6.1.7**, созданной в вашей папке.

8•. Создайте проект для вычисления длины окружности, используя константу. Сохраните проект в папке с именем **Задание 6.1.8**, созданной в вашей папке.

ПРАКТИЧЕСКАЯ РАБОТА № 10

«Составление и исполнение линейных алгоритмов обработки величин»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте окно среды **Lazarus**.
2. Разместите на форме кнопку и поле.
3. Установите следующие значения свойств кнопки:
   - ширина – *70 пикселей*;
   - высота – *30 пикселей*;
   - отступ от левой границы формы – *150 пикселей*;
   - отступ от верхней границы формы – *200 пикселей*;
   - текст на кнопке – *Свойства поля*.
4. Установите следующие значения свойств поля:
   - ширина – *200 пикселей*;
   - высота – *50 пикселей*;
   - отступ от левой границы формы – *70 пикселей*;

- отступ от верхней границы формы – *100 пикселей*;
- текст в поле – *название вашего класса*.

5. Сохраните проект в папке с именем **Практическая 10_1**, созданной в вашей папке.
6. Исполните проект.
7. Закройте окно исполнения проекта.
8. Создайте обработчик события **Click** для кнопки, исполнение которого установит красный цвет фона поля, отступ поля от верхней границы окна 150 пикселей, уменьшит отступ поля от левой границы окна на 50 пикселей, установит цвет текста красный и выведет в поле текст «Мы используем свойства поля!».
9. Сохраните новую версию проекта.
10. Исполните новую версию проекта.
11. Закройте окно исполнения проекта.
12. Создайте новый проект для вычисления значения выражения $a \cdot (4b - 3c)$.
13. Сохраните проект в папке с именем **Практическая 10_2**, созданной в вашей папке.
14. Выполните проект.
15. Закройте окно исполнения проекта.
16. Закройте окно среды **Lazarus**.

6.2. НАСТРОЙКА ПРОЕКТА

1. Каково назначение компилятора? Каковы особенности его работы?
2. Какие ошибки называют синтаксическими? Как они обнаруживаются на этапе компиляции проекта?
3. Как открыть ранее созданный проект?

ТЕСТИРОВАНИЕ ПРОЕКТА

Вы уже знаете, что при создании проекта могут возникать синтаксические ошибки. Синтаксические ошибки, которые не заметил автор проекта, находит компилятор на этапе компиляции при запуске проекта на исполнение. Необходимо исправить все такие ошибки и снова запустить на исполнение новую версию проекта.

Если исполнение проекта неожиданно прерывается или результат его исполнения неправильный, то говорят, что проект содержит **логические ошибки**.

Одним из способов выявления логической ошибки является создание **тестовых наборов** входных данных. Для каждого из этих наборов данных определяют ожидаемый результат исполнения проекта и сравнивают его с реальным результатом исполнения проекта. Если они отличаются, следует найти команды в проекте, исполнение которых является причиной появления этих различий.

Процесс поиска логических ошибок в проекте с использованием тестовых наборов входных данных называют **тестированием проекта**.

Объясним этот способ на следующем примере. Пусть необходимо составить проект для вычисления значений функции $y = \dfrac{2x - 5}{x^2 + 1}$. Процедура для

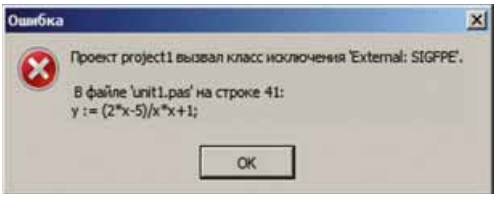


Рис. 6.8. Окно с сообщением об ошибке

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);
  y := (2*x-5)/x*x+1;
  Label1.Caption := FloatToStr(y);

end;
```

Рис. 6.9. Текст процедуры с окрашенной строкой, содержащей ошибку

вычисления значений этой функции, в которой значение переменной x вводится в поле, будет выглядеть так:

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);
  y := (2*x-5)/(x*x+1);
  Label1.Caption := FloatToStr(y);
end;
```

Часто ученики, записывая команду присваивания для вычисления значения соответствующего выражения $y := (2*x - 5)/(x*x + 1)$, забывают заключить знаменатель (а то и числитель) в скобки, то есть вводят, например, команду $y := (2*x - 5)/x*x + 1$ или $y := 2*x - 5/x*x + 1$. Такую логическую ошибку несложно заметить, применив удачную подборку тестов. Так, при тестировании для $x = 0$ станет очевидным, что проект не выдает ожидаемого правильного результата -5, а неожиданно прерывает свое исполнение, открывая окно с сообщением об ошибке (рис. 6.8).

В этом окне, кроме сообщения об ошибке, указывается номер и содержание строки с ошибкой. После выбора кнопки **ОК** это окно закрывается, а строка с ошибкой в тексте процедуры становится окрашенной (рис. 6.9).

Чтобы прервать исполнение проекта в этом случае, необходимо нажать комбинацию клавиш **Ctrl** + **F2** или выбрать кнопку **Остановить**. Далее следует понять, в чем заключается ошибка, исправить ее и снова запустить проект на исполнение. В рассматриваемом случае это происходит потому, что в ходе исполнения проекта с этим значением x должна выполниться операция деления на 0, которую выполнить невозможно.

Если как тестовый пример будет взято другое значение x, например 1, то исполнение проекта не прерывается, окно с сообщением об ошибке не открывается, но как результат будет выведено число 6, хотя ожидаемым правильным результатом является число $-3{,}5$. В таком случае также следует внимательно просмотреть текст процедуры, обнаружить ошибку и исправить ее.

Для тех, кто хочет знать больше

Часто для облегчения поиска логических ошибок в проект добавляют **комментарии**. **Комментарий** – это текст, который игнорируется компилятором и включается в текст проекта с целью облегчения его понимания и поиска логических ошибок.

Чтобы компилятор отличил команды от комментариев, комментарии следует заключать в фигурные скобки, например **{Команда вычисления значения y}** (рис. 6.10). Иногда комментарий вместо фигурных скобок заключают в круглые скобки со звездочками, например **(*Команда вывода значения y*)**.

Если текст комментария короткий и не занимает более одной строки, то можно вместо скобок поставить перед ним две косые черты, например **//Команда ввода значения x**.

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);//Команда ввода значения x
  (Команда вычисления значения y)
  y := (2*x-5)/x*x+1;
  (*Команда вывода значения y*)
  Label1.Caption := FloatToStr(y);

end;
```

Рис. 6.10. Текст процедуры с логической ошибкой и комментариями

Если комментарий занимает не более одной строки, то он может находиться после команды, а может начинаться с начала строки. Любой комментарий не может находиться внутри команды.

ПОШАГОВОЕ ИСПОЛНЕНИЕ ПРОЕКТА

Еще одним способом выявления в проекте логических ошибок является его **пошаговое исполнение**, или **трассировка. Шаг исполнения проекта** – это исполнение команд, записанных в одной строке проекта. В строке может быть записана одна команда или несколько.

Рассмотрим проект для вычисления значения дроби $z = \dfrac{x^2 - y^2}{x^2 + y^2}$. Соответствующая процедура для вычисления значения этой дроби и форма с компонентами представлены на рисунке 6.11. Для лучшего понимания процесса пошагового исполнения проекта в процедуре значение числителя и знаменателя дроби вычислены отдельными командами.

Чтобы быстрее перейти к пошаговому исполнению приведенной процедуры, целесообразно исполнить часть проекта, предшествующую исполнению этой процедуры, в автоматическом режиме. Для этого следует установить **точку остановки (точку останова)** в первой строке процедуры (рис. 6.12) и запустить проект на исполнение. Если в некоторой строке

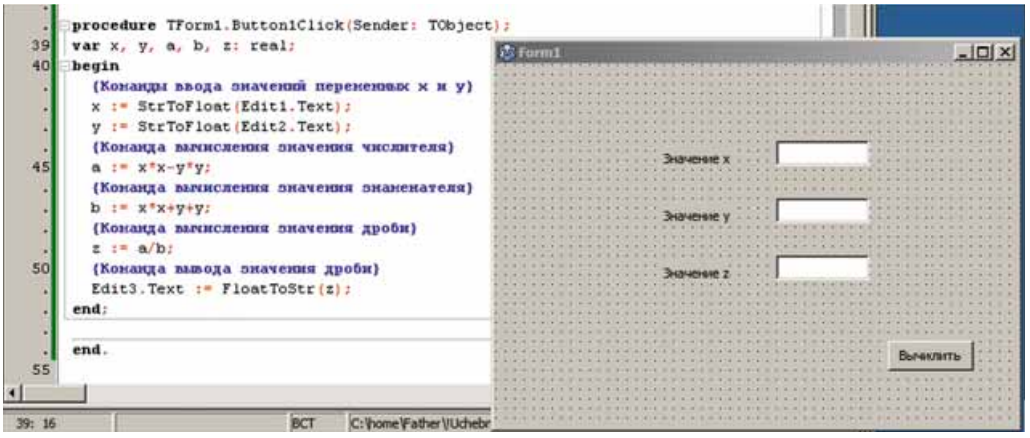


Рис. 6.11. Проект для вычисления значения дроби $z = \dfrac{x^2 - y^2}{x^2 + y^2}$

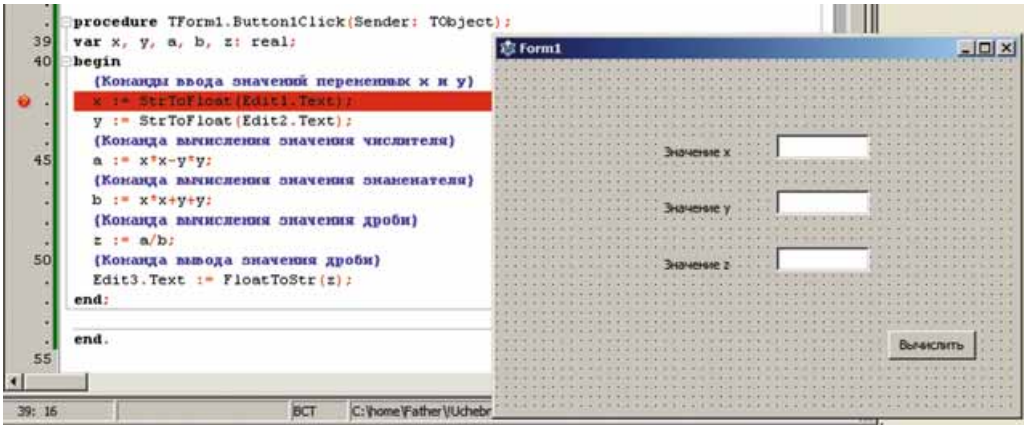


Рис. 6.12. Точка останова

проекта установлена точка останова, то после запуска проекта на исполнение в автоматическом режиме исполнится часть проекта с самого начала и до этой строки, не исполняя команды именно этой строки.

Для установки точки останова в нужной строке следует выбрать номер этой строки в тексте проекта. После этого фон выбранной строки будет окрашен красным цветом.

После запуска проекта на исполнение следует в открывшемся окне ввести в соответствующие поля значения переменных x и y и выбрать кнопку **Вычислить**. После этого начинается исполнение приведенной процедуры. Но поскольку в первой команде этой процедуры установлена точка останова, то исполнение процедуры приостановится и можно начинать ее пошаговое исполнение. Для исполнения одного шага проекта следует выполнить ***Запуск*** ⇒ ***Шаг в обход*** или нажать клавишу **F8**. При пошаговом исполнении фон строки проекта, команды которого будут исполнены при следующем шаге исполнения, окрашивается в серый цвет (рис. 6.13).

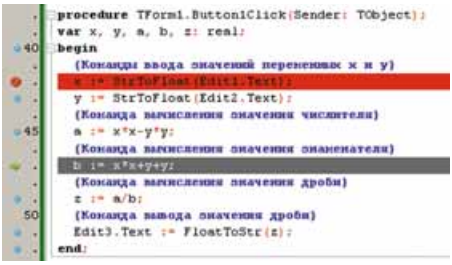


Рис. 6.13. Пошаговое исполнение

В любой момент пошаговое исполнение проекта можно прервать, нажав комбинацию клавиш **Ctrl** + **F2**, или выбрав кнопку **Остановить**, или выбрав кнопку **Запустить**. В первых двух случаях исполнение проекта прерывается, а в третьем – исполнение проекта продолжается в автоматическом режиме.

ПРОСМОТР ЗНАЧЕНИЙ ПЕРЕМЕННЫХ ПРИ ТРАССИРОВКЕ

Пошаговое исполнение проекта не имеет никакого смысла, если в ходе такого исполнения не видеть и не наблюдать, как изменяются значения переменных. Именно такое наблюдение может помочь нам понять, какие команды являются причиной логической ошибки.

Чтобы в ходе пошагового исполнения проекта наблюдать за изменением значений переменных, следует имена этих переменных поместить в окно **Список наблюдений**. Это можно сделать, используя окно **Свойства наблюдения**.

Чтобы поместить имя переменной в окно **Свойства наблюдения**, следует нажать комбинацию клавиш **Ctrl** + **F5** или выполнить ***Запуск*** ⇒ ***Добавить наблюдение*** (рис. 6.14). Имя переменной следует ввести в поле **Выражение** этого окна и выбрать кнопку **ОК**.



Рис. 6.14. Окно **Свойства наблюдения**

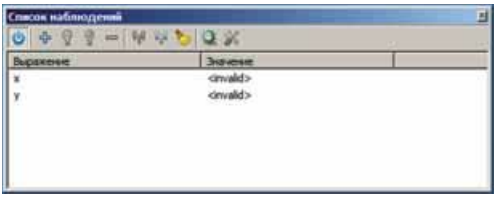


Рис. 6.15. Окно **Список наблюдений**

После этого открывается окно **Список наблюдений** (рис. 6.15) с внесенным в него именем переменной. Таким образом в это окно можно внести имена всех переменных, по значениям которых целесообразно наблюдать в ходе пошагового исполнения программы. Выбор кнопки **Добавить** этого окна также открывает окно **Свойства наблюдения**, выбор кнопки **Удалить** удаляет из этого окна выделенную строку с именем переменной и ее значением, а выбор кнопки **Удалить все** полностью удаляет список переменных, по значениям которых ведется наблюдение.

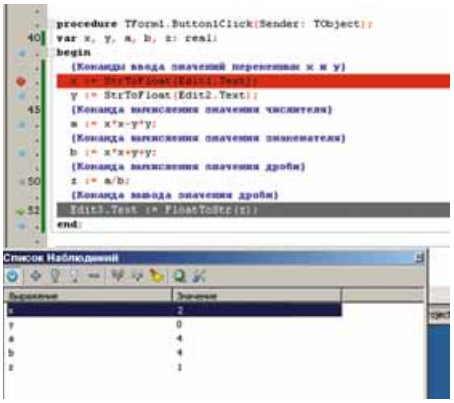


Рис. 6.16. Значения переменных в окне **Список наблюдений** в ходе пошагового исполнения проекта

До начала исполнения проекта значения переменных, используемых в проекте, не определены. Поэтому в окне **Список наблюдений** мы видим значения этих переменных **invalid** (англ. *invalid* – недействительный, несуществующий).

В ходе пошагового исполнения проекта значения этих переменных будут изменяться, что и можно наблюдать в окне **Список наблюдений** (рис. 6.16). Внимательно следя за этими изменениями и сравнивая эти значения с ожидаемыми, можно заметить, в момент исполнения какой команды проекта значение некоторой переменной становится неправильным. Именно эта команда и может быть причиной логической ошибки.

Самое важное в этом пункте

Логическая ошибка – это ошибка в проекте, которая приводит к неожиданному прерыванию его исполнения или к неправильному результату его исполнения.

Тестовый набор входных данных – это набор значений входных данных, для которых можно заранее определить правильный результат исполнения проекта, после чего сравнить его с результатом исполнения проекта для этих значений входных данных.

Тестирование проекта – это процесс поиска логических ошибок в проекте с использованием тестовых наборов входных данных.

Комментарий – это текст, который игнорируется компилятором и включается в текст проекта с целью облегчения его понимания и поиска логических ошибок. Комментарий следует заключить в фигурные скобки **{}**, или в круглые скобки со звездочками **(* *)**, или отделить от команды двумя косыми чертами **//**.

Точка остановки (точка останова) – строка в проекте, перед исполнением которой исполнение проекта приостанавливается.

Шаг исполнения проекта – это исполнение команд, записанных в одной строке проекта. В строке может быть записана одна команда или несколько.

Для исполнения одного шага проекта следует выполнить ***Запуск*** ⇒ ***Шаг в обход*** или нажать клавишу **F8**.

В любой момент пошаговое исполнение проекта можно прервать, нажав комбинацию клавиш **Ctrl** + **F2**, или выбрав кнопку **Остановить**, или выбрав кнопку **Запустить**.

Чтобы в ходе пошагового исполнения проекта следить за изменением значений переменных, следует имена этих переменных поместить в окно **Список наблюдений**. Это можно сделать, используя окно **Свойства наблюдений**.

Чтобы поместить имя переменной в окно **Свойства наблюдений**, следует нажать комбинацию клавиш **Ctrl** + **F5** или выполнить ***Запуск*** ⇒ ***Добавить наблюдение***. Имя переменной следует ввести в поле **Выражение** этого окна и выбрать кнопку **ОК**.

Дайте ответы на вопросы

1°. В чем заключается процесс тестирования проекта?
2•. Для чего выполняется тестирование проекта?
3°. Что такое комментарий в проекте? Как комментарий включить в текст проекта?
4•. Для чего в текст проекта включают комментарии?
5°. Для чего проект исполняют пошагово?
6•. В чем заключается процесс пошагового исполнения проекта?
7•. Что такое шаг исполнения проекта? Как осуществить его исполнение?
8•. Что такое точка останова? Для чего она используется?
9•. Как прервать режим пошагового исполнения проекта?
10•. Как просмотреть значения переменных при пошаговом исполнении проекта?

Выполните задания

1•. Откройте проект, например из папки **Глава 6\Задание 6.2\Проект 6.2.1**, предназначенный для решения задачи: «*Два поезда выехали одновременно навстречу друг другу. Определить, через какое время они встретятся, если известно расстояние между ними в момент*

начала движения и скорость каждого поезда». Исполнение обработчика события **Click** для кнопки должно привести к выводу искомого времени, если значения расстояния и скорости вводятся в поля. Для заданных входных данных вычислите результаты и запишите в тетрадь: а) расстояние – 1000 км, скорости – 120 км/ч и 130 км/ч; б) расстояние – 600 км, скорости – 80 км/ч и 120 км/ч; в) расстояние – 1200 км, скорости – 115 км/ч и 85 км/ч.

Исполните проект для приведенных наборов тестовых данных. Проверьте, совпадают ли результаты исполнения проекта с предварительно вычисленными результатами. Объясните, в чем может быть ошибка.



2•. Создайте три тестовых набора входных данных для тестирования проекта, предназначенного для решения задачи: «*Два поезда выехали одновременно из одной станции в противоположных направлениях. Необходимо найти расстояние, на котором они будут друг от друга через заданное время, если известна скорость каждого поезда*». Вычислите ожидаемые результаты по составленным тестам. Откройте проект, например из папки **Глава 6\Задание 6.2\Проект 6.2.2**. Протестируйте его на ваших тестовых данных. Сделайте вывод, содержит ли проект логические ошибки. Если да, то в чем может быть ошибка.

3*. Откройте проект, например из папки **Глава 6\Задание 6.2\Проект 6.2.3**. Просмотрите текст проекта и на основе анализа комментариев сделайте предположение, какую задачу решает этот проект. Опираясь на ваше предположение, составьте три тестовых набора входных данных для тестирования проекта. Вычислите ожидаемые результаты. Протестируйте проект на ваших тестовых данных. Сделайте вывод, правильным ли было ваше предположение.

4°. Откройте проект, например из папки **Глава 6\Задание 6.2\Проект 6.2.4**. Исполните проект в пошаговом режиме. Обратите внимание на изменение значений переменных в ходе исполнения проекта.



5•. Откройте проект, например из папки **Глава 6\Задание 6.2\Проект 6.2.5**. В ходе пошагового исполнения проекта проследите за изменением значений переменных ***s1*** и ***s2***.

6•. Ученики на праздник купили конфеты и печенье. Известны цены конфет и печенья и вес купленных сладостей каждого вида. Создайте проект для нахождения полной стоимости покупки. Сохраните проект в папке с именем **Задание 6.2.6**, созданной в вашей папке. Исполните проект в пошаговом режиме, проследите за значениями переменных, содержащих значения стоимости каждого товара.

7•. Откройте проект, например из папки **Глава 6\Задание 6.2\Проект 6.2.7**. В ходе пошагового исполнения проекта проследите за изменением значений переменных ***x***, ***a*** и ***s***. Выясните, как значения, которые последовательно принимает переменная ***a***, связаны со значениями переменной ***x***. Измените проект так, чтобы в результате исполнения переменная ***s*** содержала сумму $x + x^2 + x^4$ для введенного значения ***x***. Добавьте в текст программы комментарий, в котором укажите, какие изменения были внесены в программный код. Сохраните проект в папке с именем **Задание 6.2.7**, созданной в вашей папке.

8*. Откройте проект, например из папки **Глава 6\Задание 6.2\Проект 6.2.8**. Этот проект предназначен для поиска, на сколько больше времени потратил один из поездов на путь между городами, чем другой, если задано расстояние между городами и скорости каждого из поездов. Выясните, содержит ли проект логические ошибки. Если да, определите, какие команды их вызвали, и исправьте их. Запишите тестовые наборы данных, в которых вы обнаружили логические ошибки. Сохраните исправленный проект в папке с именем **Задание 6.2.8**, в вашей папке.

ПРАКТИЧЕСКАЯ РАБОТА № 11

«Настройка проекта»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте окно среды **Lazarus**.
2. Откройте проект для определения, во сколько раз произведение двух заданных действительных чисел больше суммы этих чисел. Например, проект из папки **Глава 6\Практическая 11**.
3. Создайте три тестовых набора входных данных для тестирования проекта. Запишите их в тетрадь. Вычислите ожидаемые результаты и запишите их. Протестируйте проект на ваших тестовых данных. Сравните результаты исполнения проекта с ожидаемыми результатами. Запишите вывод.
4. Добавьте комментарии для команд, в которых происходит вычисление суммы и произведения.
5. Исполните проект в пошаговом режиме. В ходе пошагового исполнения проекта следите за значениями суммы и произведения. Запишите, какие сообщения отображаются в столбце **Значение** для указанных переменных перед началом пошагового исполнения проекта, в ходе исполнения для одного из наборов входных данных и после завершения исполнения.
6. Сохраните проект в папке с именем **Практическая 11**, созданной в собственной папке.

6.3. ЛОГИЧЕСКИЕ ВЫРАЖЕНИЯ. ПЕРЕМЕННЫЕ ЛОГИЧЕСКОГО ТИПА. ЛОГИЧЕСКИЕ ОПЕРАЦИИ

1. Что такое высказывание? Какое высказывание считается истинным, а какое – ложным?
2. Какие свойства переменной вы знаете?
3. Что определяет тип переменной? Какие типы переменной вы знаете?

ВЫСКАЗЫВАНИЕ КАК ЛОГИЧЕСКОЕ ВЫРАЖЕНИЕ

Вы уже знаете, что **высказывание** – это предложение, которое содержит утверждение об определенном объекте или о связи между объектами и о котором можно однозначно сказать, **истинное** оно или **ложное**. Примером

истинного высказывания является, например, высказывание «*Киев – столица Украины*», а примером ложного – высказывание «*3 > 5*».

Основным свойством высказывания является его **истинность**. Если высказывание истинное, то считают, что значение его свойства истинность равно **true** (англ. *true* – правда). Если высказывание ложное, то считают, что значение его свойства истинность равно **false** (англ. *false* – ложь).

То есть значение свойства истинность высказываний: «*Киев – столица Украины*», «*2 * 5 – 4 = 6*», «*4 < 12*» равно **true**, значение свойства истинность высказываний: «*Этот учебник для учеников седьмого класса*», «*2 * 7 + 3 = 12*», «*3 > 5*» равно **false**.

Логика (греч. λόγος – слово, смысл, мысль, язык) – наука о формах, методах и законах правильного мышления, о способах рассуждения.

Высказывание можно рассматривать как **логическое выражение**.

Логическим выражением называют выражение, которое может принимать одно из двух значений: **true** или **false**.

Если высказывание истинные, то считают, что значение соответствующего логического выражения равно **true**. То есть, «*Киев – столица Украины*» = **true**, (*2 * 5 – 4 = 6*) = **true**, (*4 < 12*) = **true**.

Если высказывание ложное, то считают, что значение соответствующего логического выражения равно **false**. То есть «*Это учебник для учеников седьмого класса*» = **false**, (*2 * 7 + 3 = 12*) = **false**, (*3 > 5*) = **false**.

ПЕРЕМЕННЫЕ ЛОГИЧЕСКОГО ТИПА

Значение логического выражения можно присваивать некоторой переменной. То есть можно использовать следующие команды присваивания: ***x* := *2 + 12/3 = 6***, ***y* := *32 < 13***. Тогда значение переменной ***x*** равно **true**, а значение переменной ***y*** равно **false**.

Переменные, которые могут принимать одно из двух значений **true** или **false**, называют **переменными логического типа**.

Логический тип переменных обозначается **boolean**. Например, **var x, y: boolean**.

Часто для переменных логического типа используются команды **x := true** или **x := false**.

Переменные логического типа тоже считаются логическими выражениями.

ОПЕРАЦИИ НАД ЛОГИЧЕСКИМИ ВЫРАЖЕНИЯМИ

Над логическими выражениями можно выполнять **логические операции: отрицание, конъюнкция** (лат. *conjunctio* – объединение), **дизъюнкция** (лат. *disjunctio* – разъединение) и др.

Отрицанием логического выражения ***x*** называют логическое выражение, значение которого равно **true**, если значение логического выражения ***x*** равно **false**, и равно **false**, если значение логического выражения ***x*** равно **true**.

Отрицание логического выражения ***x*** обозначают **not *x***.

x	not *x*
true	false
false	true

Таблица истинности для отрицания

То есть если ***x*** = **true**, то **not** ***x*** = **false**, и если ***x*** = **false**, то **not** ***x*** = **true**. Можно составить такую таблицу, которую называют **таблицей истинности** для операции отрицания.

Для образования отрицания высказывания часто используют частицу **не**. Например, отрицанием высказывания *A* = «Существует наибольшее простое число» является высказывание **not** *A* = «Не существует наибольшего простого числа».

Иногда отрицание логического выражения ***x*** обозначают $\bar{x}$ или $\neg x$.

Конъюнкцией двух логических выражений ***x*** и ***y*** называют логическое выражение, значение которого равно **true**, если значение каждого из логических выражений ***x*** и ***y*** равно **true**; и равно **false**, если значение хотя бы одного из логических выражений ***x*** или ***y*** равно **false**.

x	***y***	***x and y***
true	**true**	**true**
true	**false**	**false**
false	**true**	**false**
false	**false**	**false**

Таблица истинности для конъюнкции

Конъюнкция двух логических выражений ***x*** и ***y*** обозначается ***x*** **and** ***y*** (англ. *and* – и).

Для образования конъюнкции двух высказываний часто используют союз ***и***. Например, высказывание «*Число 36 кратно 3 и число 36 кратно 2*» является конъюнкцией двух высказываний: «*Число 36 кратно 3*» и «*Число 36 кратно 2*».

Иногда конъюнкция двух логических выражений ***x*** и ***y*** обозначается $x \wedge y$.

Дизъюнкцией двух логических выражений ***x*** и ***y*** называют логическое выражение, значение которого равно **true**, если значение хотя бы одного из логических выражений ***x*** или ***y*** равно **true**; и равно **false**, если значение каждого из логических выражений ***x*** и ***y*** равно **false**.

Дизъюнкция двух логических выражений ***x*** и ***y*** обозначается ***x*** **or** ***y*** (англ. *or* – или).

Для образования дизъюнкции высказываний часто используют союз **или**. Например, высказывание «*11* ⩽ *21*» (11 меньше или равно 21) является дизъюнкцией двух высказываний: «*11* < *21*» и «*11* = *21*».

x	***y***	***x or y***
true	**true**	**true**
true	**false**	**true**
false	**true**	**true**
false	**false**	**false**

Таблица истинности для дизъюнкции

Иногда дизъюнкция двух логических выражений ***x*** и ***y*** обозначается $x \vee y$.

Часто используют логические выражения, содержащие несколько логических операций. Например: **(x or y) and (x or z)**.

В таких выражениях для логических операций, рассмотренных выше, определен **приоритет выполнения операций**: сначала выполняются операции отрицания, затем – операции конъюнкции, затем – операции дизъюнкции. Для изменения этого порядка выполнения логических операций используют скобки.

Для вычисления значений логического выражения можно использовать таблицу истинности. В ней необходимо рассмотреть все возможные значения переменных логического типа, входящих в это выражение.

x	y	z	1	2	3
			x or y	x or z	(x or y) and (x or z)
true	true	true	true	true	true
true	true	false	true	true	true
true	false	true	true	true	true
false	true	true	true	true	true
true	false	false	true	true	true
false	true	false	true	false	false
false	false	true	false	true	false
false	false	false	false	false	false

Интересные факты из истории

Определение операций над переменными логического типа и свойства этих операций сформулировал английский математик и философ **Джордж Буль** (1815–1864) (рис. 6.17). В 1854 г. вышла его основная работа «Исследования законов мышления, на которых основаны математические теории логики и вероятностей». В ней исследуется система, которую сегодня называют «алгеброй высказываний», или «булевой логикой». Булевая логика стала основным математическим инструментом для создания компьютеров.



Рис. 6.17. Джордж Буль

Самое важное в этом пункте

Логическим выражением называют выражение, которое может принимать одно из двух значений: **true** или **false.**

Высказывание является примером **логического выражения.** Если высказывание **истинное,** то считают, что значение соответствующего логического выражения равно **true.** Если высказывание **ложное,** то считают, что значение соответствующего логического выражения равно **false.**

Переменные, которые могут принимать одно из двух значений **true** или **false,** называют **переменными логического типа.** Логический тип переменных обозначается **boolean.** Переменные логического типа также считаются логическими выражениями.

x	not x
true	false
false	true

Таблица истинности для отрицания

x	y	x and y
true	true	true
true	false	false
false	true	false
false	false	false

Таблица истинности для конъюнкции

x	y	x or y
true	true	true
true	false	true
false	true	true
false	false	false

Таблица истинности для дизъюнкции

В логических выражениях для логических операций, рассмотренных выше, определен следующий **приоритет выполнения операций**: сначала выполняются операции отрицания, затем – операции конъюнкции, потом – операции дизъюнкции. Для изменения этого порядка выполнения логических операций используют скобки.

Для вычисления значений логического выражения можно использовать таблицу истинности. В ней необходимо рассмотреть все возможные значения переменных логического типа, входящих в это выражение.

Дайте ответы на вопросы

1°. Какое выражение называют логическим? Приведите примеры.
2°. Какие переменные называют переменными логического типа?
3°. Как описываются переменные логического типа?
4•. Что такое отрицание логического выражения?
5•. Что такое конъюнкция двух логических выражений?
6•. Что такое дизъюнкция двух логических выражений?
7°. Каков приоритет выполнения логических операций?
8*. Что изучает наука логика?

Выполните задания

1°. Укажите порядок выполнения логических операций в логических выражениях:
а) **a or not a and b;**
б) **not a or not b and a;**
в) **(a or not b) and (not a or not b);**
г) **not a and not b or c and d.**

2•. Составьте таблицы истинности логических выражений с переменными логического типа ***a*** и ***b***:
а) **a or a and b;**
б) **a or not a and b;**
в) **a or b or not b;**
г) **not a or not b;**
д) **(a or not b) and (a or not b);**
е) **a and not b or not a and b.**

3•. Сформулируйте отрицание высказываний и определите их истинность:
а) Число 3 является делителем числа 545.
б) Автомобиль не имеет права ехать на красный свет.
в) Существуют параллелограммы с прямыми углами.
г) Уравнение $2x^2 - 3x + 1 = 0$ имеет целый корень.
д) Не существует натурального числа, которое делится на 2.
е) Существует целое число, которое делится на все целые числа.

4•. Среди указанных высказываний найдите конъюнкции и дизъюнкции и определите их истинность:
а) Число 27 кратно 3 и кратно 9.
б) $17 < 42 < 18$.
в) Число 2 простое или четное.
г) Треугольник ABC является остроугольным, прямоугольным или тупоугольным.
д) Диагонали любого параллелограмма перпендикулярны и точкой пересечения делятся пополам.
е) $7^2 = 49$ и $(-7)^2 = 49$.
ж) $21 \leqslant 21$.
з) $21 \leqslant 18$.

5•. Пусть A = «Сергиенко изучает английский язык», B = «Сергиенко имеет оценку 8 по информатике». Сформулируйте высказывания:

а) **A and not B;** в) **A or not A and B;**

б) **not A or B;** г) **not A and not B and A.**

6*. Определите истинность или ложность высказываний для приведенных значений переменных:

1. **«$x > 0$ и $y > 0$ или $x < 0$ и $y < 0$»**
2. **«$x > 0$ и не $y < 0$ или $x < 0$ и $y > 0$»**

Значения переменных:

а) $x = 5$, $y = 8$;

б) $x = 5$, $y = -8$;

в) $x = -5$, $y = 8$;

г) $x = -5$, $y = -8$.

6.4. АЛГОРИТМЫ С ВЕТВЛЕНИЯМИ

1. Какие выражения называют логичными? Какие значения они могут принимать?
2. Назовите логические операции. Приведите таблицу истинности каждой из них.
3. Что такое ветвление? Какие виды ветвления вам известны? Чем они отличаются друг от друга?

АЛГОРИТМЫ С ВЕТВЛЕНИЕМ

Вы уже знаете, что алгоритмы могут содержать **ветвления.** Вы также знаете, что ветвление как фрагмент алгоритма начинается с команды проверки условия и результатом выполнения этой команды может быть либо **истина (Да, true)**, либо **ложь (Нет, false).** И в зависимости от результата исполнения этой команды будут исполняться те или иные команды.

В общем случае в этой команде определяется значение определенного логического выражения, которое может быть либо **true**, либо **false.**

В общем случае блок-схемы ветвлений имеют следующий вид (рис. 6.18 и 6.19):

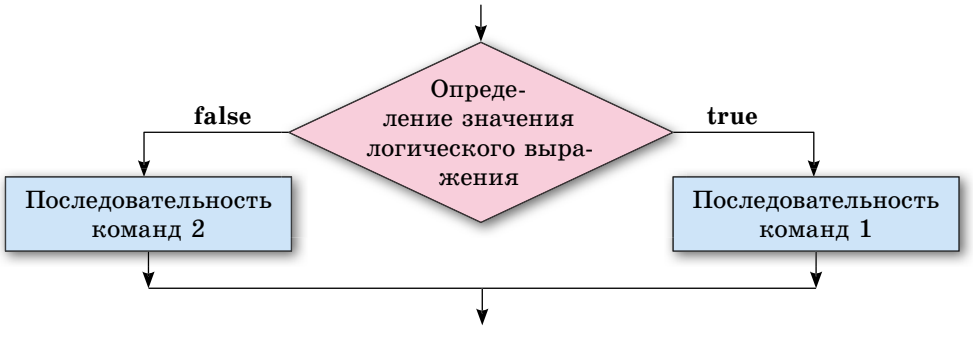


Рис. 6.18. Полное ветвление

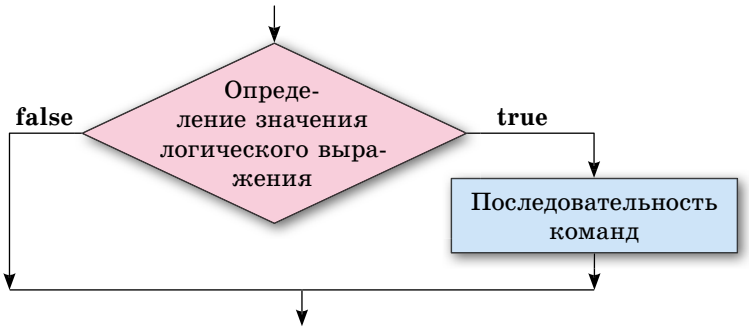


Рис. 6.19. Неполное ветвление

ВЕТВЛЕНИЯ В OBJECT PASCAL

Команда полного ветвления в языке программирования **Object Pascal** имеет такой общий вид:

```
If<логическое выражение>
Then begin
        <последовательность команд 1>
     end
Else begin
        <последовательность команд 2>
     end;
```

(англ. *if* – если, *then* – то, тогда, *else* – иначе).

Исполняется эта команда так: вычисляется значение логического выражения; если это значение **true**, то исполняется **последовательность команд 1** и после этого исполняется команда, следующая за ветвлением; если это значение **false**, то исполняется **последовательность команд 2** и после этого исполняется команда, следующая за ветвлением.

Есть в **Object Pascal** и команда неполного ветвления:

```
If<логическое выражение>
Then begin
        <последовательность команд>
     end;
```

Исполняется эта команда так: вычисляется значение логического выражения; если это значение **true**, то исполняется **последовательность команд** и после этого исполняется команда, следующая за ветвлением; если это значение **false**, сразу исполняется команда, следующая за ветвлением.

Обращаем ваше внимание:

- если после ключевых слов **Then** и **Else** следует только по одной команде, то операторные скобки **begin** и **end** можно не ставить;
- перед **Else** ставить точку с запятой нельзя.

Рассмотрим задачу, алгоритм решения которой содержит ветвление.

Задача 1. Вычислить значение выражения $(a + 2b) / (a - 3b)$, где a, b – действительные числа.

Поскольку выражение содержит действие деления на выражение, которое может быть равно нулю (например, при $a = 3$ и $b = 1$), алгоритм реше-

ния этой задачи должен содержать ветвление, в котором будут учтены два возможных случая: значение выражения $a - 3b$ равно 0 и значение выражения $a - 3b$ не равно 0.

Блок-схема алгоритма решения этой задачи будет выглядеть так (рис. 6.20):

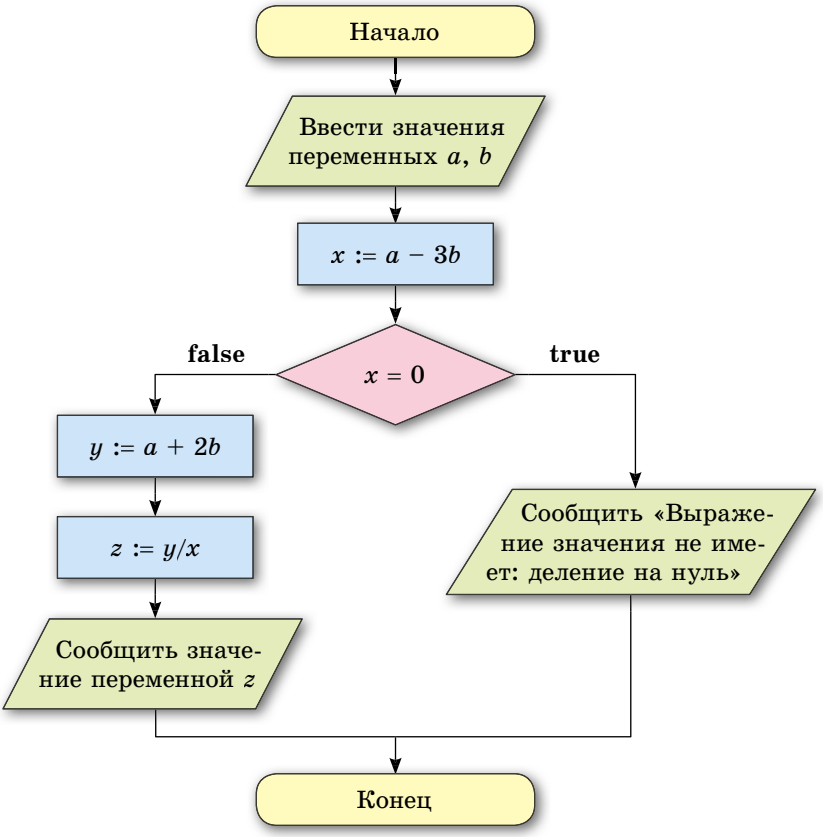


Рис. 6.20. Блок-схема алгоритма решения задачи 1

Если записать на языке **Object Pascal** алгоритм, приведенный на рисунке 6.20, то он будет выглядеть так:

```
a := StrToFloat(Edit1.Text);
b := StrToFloat(Edit2.Text);
x := a–3*b;
If x = 0
Then Label1.Caption := 'Выражение значения не имеет: деление на нуль'
Else begin
        y := a+2*b;
        z := y/x;
        Label1.Caption := FloatToStr(z);
     end;
```

Обращаем ваше внимание, что вычисление значения выражения $a - 3b$ можно было бы не выполнять в отдельной команде и не запоминать как значение переменной x, а выполнять непосредственно в команде ветвле-

ния. Также вычисление значения выражения $(a + 2b) / (a - 3b)$ можно записать не двумя командами, а одной. Тогда этот фрагмент проекта будет выглядеть так:

```
a := StrToFloat(Edit1.Text);
b := StrToFloat(Edit2.Text);
If a–3*b = 0
Then Label1.Caption := 'Выражение значения не имеет: деление на нуль'
Else begin
        z := (a+2*b) /(a–3*b);
        Label1.Caption := FloatToStr(z);
     end;
```

Рассмотрим задачу, при решении которой в команде ветвления используется конъюнкция.

Задача 2. Вычислить значение заданной функции

$$y = \begin{cases} 2x - 12, \text{ если } -3 < x < 5, \\ 7 - 8x, \text{ для всех других } x. \end{cases}$$

Фрагмент проекта для решения этой задачи будет выглядеть так:

```
x := StrToFloat(Edit1.Text);
If (x > –3) and (x < 5)
Then y := 2*x – 12
Else y := 7–8*x;
Label1.Caption := FloatToStr(y);
```

Рассмотрим, наконец, задачу, при решении которой используется более сложное ветвление.

Задача 3. Даны два числа. Определить, равны ли они. Если нет, то какое из них больше?

Блок-схема алгоритма решения этой задачи будет выглядеть так, как на рисунке 6.21.

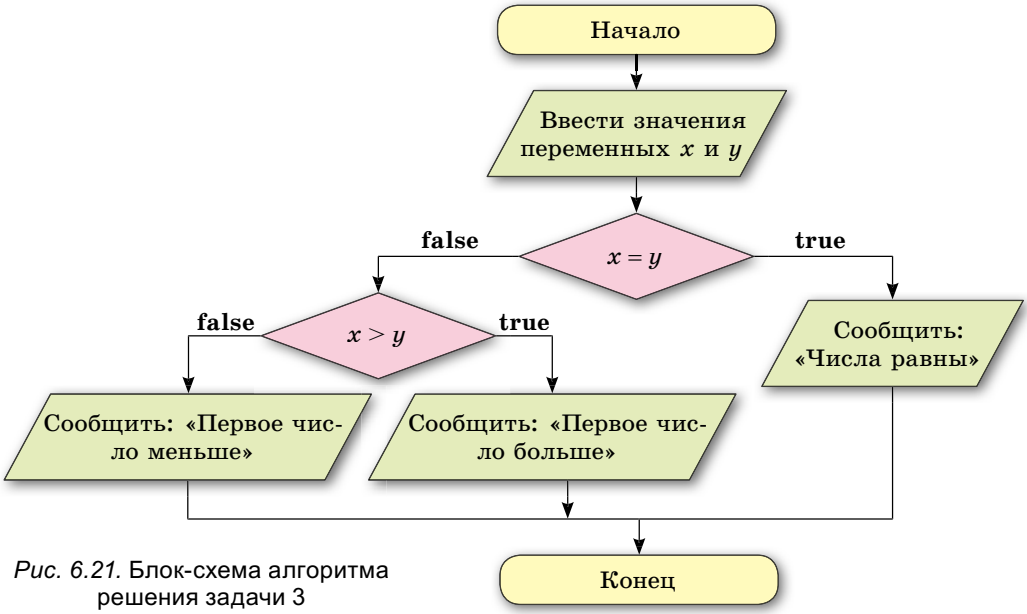


Рис. 6.21. Блок-схема алгоритма решения задачи 3

В этом алгоритме внутри одного ветвления используется другое ветвление. Вот как выглядит фрагмент соответствующего проекта:

```
If x = y
Then Label1.Caption := 'Числа равны'
Else If x>y
  Then Label1.Caption := 'Первое число больше'
  Else Label1.Caption := 'Первое число меньше';
```

Самое важное в этом пункте

В общем случае блок-схемы ветвлений будут выглядеть так (рис. 6.22 и 6.23):

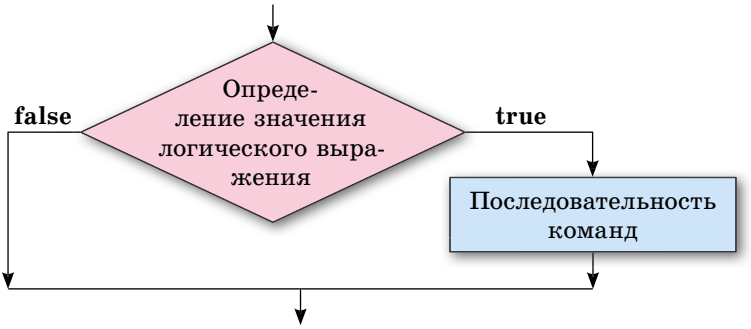


Рис. 6.23. Неполное ветвление

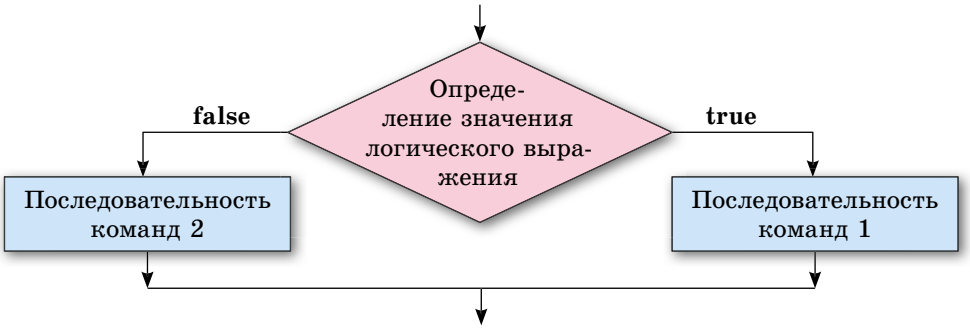


Рис. 6.22. Полное ветвление

Команда полного ветвления в языке программирования **Object Pascal** будет выглядеть так:

```
If <логическое выражение>
Then begin
        <последовательность команд 1>
     end
Else begin
        <последовательность команд 2>
     end;
```

Исполняется эта команда следующим образом: вычисляется значение логического выражения; если это значение **true**, то исполняется **последо-**

вательность команд 1 и после этого исполняется команда, следующая за ветвлением; если это значение **false**, то исполняется **последовательность команд 2** и после этого исполняется команда, следующая за ветвлением.

Есть в **Object Pascal** и команда неполного ветвления:

```
If <логическое выражение>
Then begin
        <последовательность команд>
     end;
```

Исполняется эта команда следующим образом: вычисляется значение логического выражения; если это значение **true**, то исполняется **последовательность команд** и после этого исполняется команда, следующая за ветвлением; если это значение **false**, сразу исполняется команда, следующая за ветвлением.

Дайте ответы на вопросы

1•. Как выглядит блок-схема полного ветвления? Опишите особенности исполнения такого фрагмента алгоритма.
2•. Как выглядит блок-схема неполного ветвления? Опишите особенности исполнения такого фрагмента алгоритма.
3•. В чем различия в исполнении полного и неполного ветвления?
4°. Как выглядит команда полного ветвления в **Object Pascal**?
5°. Как выглядит команда неполного ветвления в **Object Pascal**?
6•. В чем различие исполнения команд полного и неполного ветвления в **Object Pascal**?

Выполните задания

1•. Объясните ход исполнения команды:

```
If <логическое выражение 1>
Then begin
        <команды 1>
     end
Else If <логическое выражение 2>
     Then begin
             <команды 2>
          end
     Else begin
             <команды 3>
          end;
```

2•. Объясните ход исполнения последовательности команд:

```
If <логическое выражение 1>
Then begin
        <команды 1>
     end;
If <логическое выражение 2>
Then begin
        <команды 2>
     end;
```

3*. Объясните, чем исполнение команды в задании № 1 отличается от исполнения последовательности команд в задании № 2.

4•. Создайте проект для вычисления значения выражения $(a + b) - c : (a - b)$. Сохраните проект в папке с именем **Задание 6.4.4**, созданной в вашей папке.



5•. Создайте проект для вычисления значения выражения $a + b - c : (a - 2b)$. Сохраните проект в папке с именем **Задание 6.4.5**, созданной в вашей папке.

6•. Создайте проект, который определяет меньшее из двух произвольных действительных чисел или выводит сообщение, что числа равны. Сохраните проект в папке с именем **Задание 6.4.6**, созданной в вашей папке.



7•. Создайте проект, который определяет большее из двух неравных действительных чисел. Сохраните проект в папке с именем **Задание 6.4.7**, созданной в вашей папке.



8°. Создайте проект, который вычисляет модуль произвольного действительного числа. Сохраните проект в папке с именем **Задание 6.4.8**, созданной в вашей папке.

9*. Создайте проект, который определяет наименьшее число среди трех неравных чисел. Сохраните проект в папке с именем **Задание 6.4.9**, созданной в вашей папке.

10*. Приведите примеры правил по украинскому языку, математике, другим предметам, содержащих ветвления.

11*. Приведите примеры жизненных ситуаций, которые можно описать алгоритмом с ветвлением.

12*. Создайте проект, в котором кнопка будет перемещаться на 20 пикселей вправо при наведении на нее курсора. После того как кнопка исчезнет за правой границей окна, она должна появиться у левой ее границы. Сохраните проект в папке с именем **Задание 6.4.12**, созданной в вашей папке.

13*. Создайте проект для нахождения x из уравнения $ax = b$. Сохраните проект в папке с именем **Задание 6.4.13**, созданной в вашей папке.



14*. Создайте проект для нахождения x из уравнения $ax + b = c$. Исполните его при:

1) $a = 2$; $b = -8$; $c = 18$; 2) $a = 20$; $b = 5$; $c = 5$;

3) $a = 0$; $b = 12$; $c = 3$; 4) $a = 0$; $b = 10$; $c = 10$.

Сохраните проект в папке с именем **Задание 6.4.14**, созданной в вашей папке.

15*. Создайте проект для определения, можно ли из трех отрезков с заданными длинами образовать треугольник. Сохраните проект в папке с именем **Задание 6.4.15**, созданной в вашей папке.



16•. Создайте проект, в котором вычисляется значение функции

$$y = \begin{cases} 15 - 3x, \text{ если } -1 < x < 4, \\ 6 + 4x, \text{ для всех других } x. \end{cases}$$

Сохраните проект в папке с именем **Задание 6.4.16**, созданной в вашей папке.

6.5. ФЛАЖКИ. ПОЛЕ С РАСКРЫВАЮЩИМСЯ СПИСКОМ. ПЕРЕКЛЮЧАТЕЛИ

1. Какой вид команды полного и неполного ветвления в языке **Object Pascal** и как они исполняются?
2. Какие свойства метки и поля вы знаете? Как можно изменить их значения?
3. Для чего в программах, с которыми вы работали, использовались флажки, поле с раскрывающимся списком, переключатели?

ФЛАЖКИ И ИХ ИСПОЛЬЗОВАНИЕ

Рассмотрим компонент **флажок**. Значение некоторых его свойств, в частности свойства **Checked** (англ. *checked* – проверено), могут быть использованы в логическом выражении в команде ветвления.

Для расположения флажка на форме нужно дважды щелкнуть на кнопке **TCheckBox** (англ. *checkbox* – коробка для проверки) на вкладке **Standard** с компонентами. После этого флажок появится в верхней левой части формы (рис. 6.24), а его имя добавится в список компонентов проекта в поле **Компоненты** окна **Инспектор объектов**. При необходимости поле можно перетащить в любое место формы.



Рис. 6.24. Флажок на форме

Свойства флажка **Caption**, **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font**, **Width**, **Height** аналогичные соответствующим свойствам метки. В частности, значения свойств **Width** и **Height** флажка определяются длиной текста справа от флажка и размером шрифта, которым выведен этот текст. Сам этот текст является значением свойства **Caption** флажка. Этот текст, как и для метки, можно вводить непосредственно на вкладке **Свойства** в поле **Caption**, а можно вводить в окне **Диалог ввода строк**, которое открывается выбором кнопки ... справа от поля **Caption**.

Свойство **Checked** флажка может принимать одно из двух значений: **true** (отметка флажка установлена, флажок установлен) или **false** (отметка флажка не установлена, флажок снят).

Если на форме расположено несколько флажков, то среди них можно установить один или несколько, не устанавливать ни одного или установить все. Если флажок установлен, то в нем появляется отметка ☑.

Установлением отметок флажков можно задать значение одного или нескольких свойств объекта из некоторого определенного набора свойств.

Рассмотрим, как можно использовать флажки для реализации ветвления.

Пусть, например, нам необходимо в ходе исполнения проекта изменять значения либо одного из свойств окна (заголовок, цвет или размеры), либо нескольких из них, либо не изменять значение ни одного из них.

Для этого разместим на форме кнопку и три флажка (рис. 6.25). При размещении флажков на форме для каждого из них по умолчанию значение свойства **Checked** устанавливается **false**. Поскольку при исполнении проекта ни один флажок может быть не установлен, то целесообразно оставить такие начальные значения свойства **Checked** для всех флажков.

Создадим процедуру для обработки события **Click** для кнопки и в нее поместим следующие три последовательные (не вложенные друг в друга) команды ветвления:

```
If CheckBox1.Checked
Then Form1.Caption := 'Заглавие изменено';
If CheckBox2.Checked
Then Form1.Color := clRed;
If CheckBox3.Checked
Then begin
        Form1.Width := 500;
        Form1.Height := 200;
     end;
```

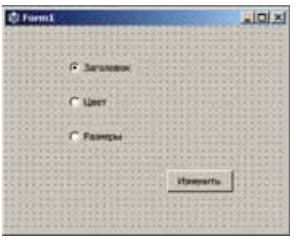


Рис. 6.25. Форма с кнопкой и тремя флажками

В каждой из этих команд, независимо друг от друга, будет определяться, установлен флажок или нет. И в зависимости от этого, в каждой из них будут или не будут изменяться значения соответствующих свойств окна.

ПОЛЕ С РАСКРЫВАЮЩИМСЯ СПИСКОМ И ЕГО ИСПОЛЬЗОВАНИЕ

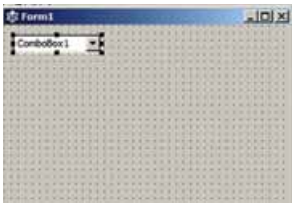


Рис. 6.26. Поле с раскрывающимся списком на форме

Еще одним компонентом, значения свойств которого можно использовать в логическом выражении в команде ветвления, является **поле с раскрывающимся списком.**

Для размещения на форме поля с раскрывающимся списком следует дважды щелкнуть на кнопке  **TComboBox** (англ. *combobox* – поле со списком) на вкладке **Standard** с компонентами. После этого поле с раскрывающимся списком появится в верхней левой части формы (рис. 6.26), а его имя добавится в список компонентов проекта в поле **Компоненты** окна **Инспектор объектов.** При необходимости поле с раскрывающимся списком можно перетащить в любое место формы.

Свойства **Top, Left, Height, Width, Name, Color, Enabled, Visible, Font, Text** поля с раскрывающимся списком полностью аналогичны соответствующим свойствам поля.

Задать содержание раскрывающегося списка можно в окне **Диалог ввода строк** (рис. 6.27), которое открывается выбором кнопки  справа от поля **Items** (англ. *items* – предметы) вкладки **Свойства**.

В ходе исполнения проекта можно открыть этот список и выбрать любую его строку. При этом соответствующий текст попадает в поле и становится значением свойства **Text** поля с раскрывающимся списком.

Рассмотрим, как можно использовать поле с раскрывающимся списком в команде ветвления.

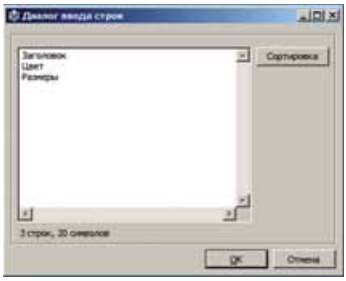


Рис. 6.27. Окно **Диалог ввода** строк

Пусть, например, в ходе исполнения проекта необходимо изменять значение только одного из свойств окна: либо заголовок, либо цвет, либо размеры.

Разместим на форме поле с раскрывающимся списком и создадим соответствующий список (рис. 6.27). Разместим также на форме кнопку и составим процедуру – обработчик события **Click** для этой кнопки, при исполнении которой будем выбирать одну из строк раскрывающегося списка (рис. 6.28). В зависимости от выбранной строки будет изменяться или заголовок формы, или ее цвет, или ее размеры.

Выберем первый пункт списка *Заголовок* и нажмем кнопку. Тогда при исполнении процедуры значение логического выражения **(ComboBox1.Text = 'Заголовок') = true**, а значит будет исполняться команда **Form1.Caption := 'Заголовок изменен'**, которая изменит заголовок окна, после чего исполнение команды ветвления и всей процедуры закончится.

Если выбрать второй пункт списка *Цвет* и нажать кнопку, то при исполнении процедуры значение логического выражения **(ComboBox1.Text = 'Заголовок') = false**. Тогда вычисляется значение логического выражения **ComboBox1.Text = 'Цвет'**, и оно равно **true**. А значит будет исполняться команда **Form1.Color:= clRed**, которая изменит цвет окна, после чего исполнение команды ветвления и всей процедуры закончится.

Если выбрать третий пункт списка *Размеры* и нажать кнопку, то при исполнении процедуры значение логического выражения **(ComboBox1.Text = 'Заголовок') = false**. После этого вычисляется значение логического выражения **ComboBox1.Text = 'Цвет'**, и оно тоже равно **false**. После этого вычисляется значение логического выражения **ComboBox1.Text = 'Размеры'**, и оно равно **true**. А значит будут исполняться команды **Form1.Width := 600** и **Form1.Height := 300**, которые изменят ширину и высоту окна, после чего исполнение команды ветвления и всей процедуры закончится.

Если не выбрать ни один пункт списка и нажать кнопку, то значение всех логических выражений в ветвлении будут **false**, и в этом случае не будут изменяться значения ни одного свойства окна, и исполнение процедуры закончится.

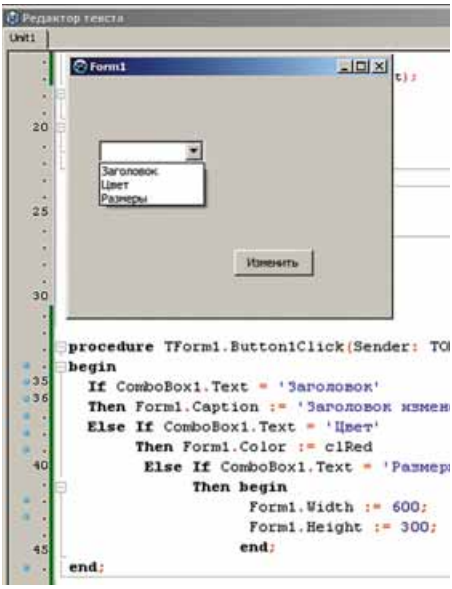


Рис. 6.28. Процедура изменения значения заголовка, цвета или размеров окна и окно исполнения проекта

Для тех, кто хочет знать больше

Переключатели и их использование

Для выбора одного варианта из нескольких возможных, например, установить для определенного свойства объекта одно значение из некоторого определенного набора значений, можно использовать **переключатели**.

Для размещения на форме переключателя следует дважды щелкнуть на кнопку **TRadioButton** (англ. *radiobutton* – круглая кнопка) на вкладке **Standard** с компонентами.

Если на форме есть несколько переключателей, то среди них можно выбрать только один. Если переключатель выбран, тогда в нем появляется отметка .

Все те свойства, которые мы рассматривали для флажка, имеет и переключатель, в том числе и свойство **Checked**, значение которого можно использовать в логическом выражении в команде ветвления.

Рассмотрим, как можно изменять значение только одного из свойств окна: либо заголовок, либо цвет, либо размеры, использовав при этом не поле с раскрывающимся списком, а переключатели.

Разместим на форме кнопку и три переключателя (рис. 6.29). При размещении переключателей на форме для каждого из них по умолчанию устанавливается значение свойства **Checked** – ***false***. До запуска проекта изменим значение этого свойства у первого переключателя на **true**. После этого на форме он будет выглядеть выбранным. Таким он будет и после запуска проекта на исполнение.

Создадим процедуру – обработчик события **Click** для кнопки и в ее текст поместим такую команду:

```
If RadioButton1.Checked
Then Form1.Caption := 'Заголовок изменен'
Else If RadioButton2.Checked
  Then Form1.Color := clRed
  Else  begin
          Form1.Width := 600;
          Form1.Height := 300;
        end;
```

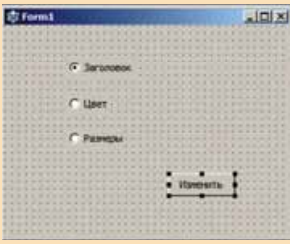


Рис. 6.29. Форма с кнопкой и тремя переключателями

Начинаться исполнение этой команды будет с определения значения свойства **RadioButton1.Checked**.

Если при наступлении события **Click** для кнопки будет выбран первый переключатель, то **RadioButton1.Checked** = **true**. Поэтому исполнится команда изменения заголовка окна, а другие части этой команды ветвления исполняться не будут.

Если при наступлении события **Click** для кнопки будет выбран второй переключатель, то **RadioButton1.Checked** = **false**. Поэтому будут исполняться команды после первого ключевого слова **Else**. Это команда ветвления, которая является частью первой команды ветвления. При ее исполнении определяется значение свойства **RadioButton2.Checked**. Поскольку будет выбран второй переключатель, то это значение равно **true**. Поэтому исполнится команда изменения цвета окна, а другие части команды исполняться не будут.

Если при наступлении события **Click** для кнопки будет выбран третий переключатель, то **RadioButton1.Checked** = **false**. Поэтому будут исполняться команды после первого ключевого слова **Else**, то есть внутренняя команда ветвления. При ее исполнении определяется значение свойства **RadioButton2.Checked**. Поскольку второй переключатель не выбран, то его значение **false**. Тогда будут ис-

полняться команды после второго ключевого слова **Else**, то есть две команды изменения размеров окна.

Если перед запуском этого проекта на исполнение ни у одного переключателя не изменять значение свойства **Checked**, установленное по умолчанию, то при запуске проекта ни один из них не будет выбран. Если и при исполнении проекта не выбран ни один из переключателей, при выборе кнопки изменятся размеры окна (подумайте, почему состоятся именно такие изменения). Чтобы этого не произошло, необходимо либо обязательно выбирать один из переключателей, либо после второго ключевого слова **Else** вставить еще одну команду ветвления:

```
If RadioButton1.Checked
Then Form1.Caption := 'Заголовок изменен'
Else If RadioButton2.Checked
     Then Form1.Color := clRed
     Else If RadioButton3.Checked
          Then begin
                 Form1.Width := 600;
                 Form1.Height := 300;
               end;
```

Самое важное в этом пункте

Значение свойства **Checked** флажков может быть использовано в логическом выражении команды ветвления. Оно равно **True**, если флажок установлен, или **False**, если флажок не установлен (снят).

Для размещения на форме флажка следует дважды щелкнуть на кнопке **TCheckBox** на вкладке **Standard** с компонентами. Если на форме расположено несколько флажков, то среди них можно установить один или несколько, не установить ни одного или установить все. Поэтому флажки используют для того, чтобы в ходе исполнения проекта выбрать или один из возможных вариантов, или несколько, или выбрать все, или не выбрать ни одного.

Для обработки состояния флажков (установлен или не установлен) в проект включают для каждого из них последовательно записанные команды ветвления, не зависимые друг от друга.

Для того чтобы в ходе исполнения проекта выбрать только один из возможных вариантов, можно использовать **поле с раскрывающимся списком**. Для размещения на форме поля с раскрывающимся списком следует дважды щелкнуть на кнопке **TComboBox** на вкладке **Standard** с компонентами.

Дайте ответы на вопросы

1•. Для чего используют флажки?
2°. Какие значения может принимать свойство **Checked** для флажка?
3•. Для чего используют поле с раскрывающимся списком?
4•. Как нужно записывать команды ветвления, в которых используется значение свойства **Checked** для флажков?
5•. Как целесообразно записать команды ветвления, в которых используются значения свойств поля с раскрывающимся списком?

6*. Каковы отличия в исполнении команд ветвления с использованием значений свойств флажков и поля с раскрывающимся списком?

7*. Для чего используют переключатели?

8*. Какие значения может принимать свойство **Checked** для переключателя?

Выполните задания

1•. Нарисуйте блок-схему и объясните ход исполнения последовательности команд:

```
If CheckBox1.Checked
Then begin
        <команды 1>
     end;
If CheckBox2.Checked
Then begin
        <команды 2>
     end;
```

2•. Нарисуйте блок-схему и объясните ход исполнения команд:

```
If ComboBox1.Text = '<Текст 1>'
Then begin
        <команды 1>
     end
Else If ComboBox1.Text = '<Текст 2>'
     Then begin
             <команды 2>
          end
     Else If ComboBox1.Text = '<Текст 3>'
          Then begin
                  <команды 3>
               end;
```

3*. Нарисуйте блок-схему и объясните ход исполнения команд:

```
If RadioButton1.Checked
Then begin
        <команды 1>
     end
Else If RadioButton2.Checked
     Then begin
             <команды 2>
          end
     Else begin
             <команды 3>
          end;
```

4*. Объясните отличия в исполнении команды ветвления из задания № 2 и последовательности двух команд ветвления из задания № 1.

5•. Создайте проект, в котором можно вычислить сумму, или разность, или произведение, или частное от деления двух произвольных действительных чисел в зависимости от выбранного пункта раскрывающегося списка. Сохраните проект в папке с именем **Задание 6.5.5**, созданной в вашей папке.

6•. Создайте проект, в котором можно вычислить сумму, разность, произведение, частное от деления двух произвольных действительных чисел в зависимости от установленных отметок флажков. Сохраните проект в папке с именем **Задание 6.5.6**, созданной в вашей папке.

7•. Рабочий изготавливает детали. Известно количество реально изготовленных деталей за месяц и стоимость одной детали. Если рабочий выполняет план, то он получает премию в размере 10 % от заработанной в результате изготовления деталей суммы. Если рабочий перевыполняет план, то его премия составляет 30 % от заработанной суммы. А если он недовыполнит план, то с него снимается 20 % от заработанной суммы. Создайте проект, определяющий сумму денег, которую заработал рабочий за месяц. Сохраните проект в папке с именем **Задание 6.5.7**, созданной в вашей папке.

8•. Рабочий имеет фиксированную заработную плату за работу в течение месяца. Если у рабочего есть дети, он получает доплату в размере 20 % от его месячной заработной оплаты. Если у рабочего в течение месяца были случаи нарушения трудовой дисциплины, с него взымают штраф в размере 15 % от его месячной заработной оплаты. Создайте проект, определяющий сумму денег, которую заработал рабочий за месяц. Сохраните проект в папке с именем **Задание 6.5.8**, созданной в вашей папке.

6.6. ЦИКЛ СО СЧЕТЧИКОМ

1. Какие процессы называют циклическими? Приведите примеры.
2. Что такое цикл в алгоритме? Что такое тело цикла?
3. Каков вид команды **Повторить N раз**? Как она исполняется?

КОМАНДА ЦИКЛА СО СЧЕТЧИКОМ В OBJECT PASCAL

В языке программирования **Object Pascal** есть несколько команд, которые могут реализовать цикл. Одна из них – **команда цикла со счетчиком**. Ее целесообразно использовать в тех случаях, когда количество повторений команд тела цикла известно еще до начала исполнения команды цикла. Общий вид этой команды такой:

```
for <имя переменной> := <выражение1> to <выражение2> do
begin
  <команды тела цикла>
end;
```

Строка **for <имя переменной> := <выражение1> to <выражение2> do** (англ. *for* – для, *to* – к, *do* – делать, исполнять) называется **строкой заголовка команды цикла со счетчиком**. Переменную в строке заголовка команды цикла со счетчиком, которая стоит перед знаком присваивания, называют **счетчиком цикла**.

Блок-схема команды цикла со счетчиком приведена на рисунке 6.30.

Счетчик цикла, **выражение1** и **выражение2** могут принимать только целые значения. Если тело цикла состоит только из одной команды, операторные скобки **begin** и **end** можно не ставить.

Исполняется команда цикла со счетчиком следующим образом:

1. Присвоить счетчику цикла ***i*** значение **выражение1.**
2. Вычислить значение логического выражения ***i*** ⩽ **выражение2.**
3. Если значение логического выражения **true**, то исполнить команды тела цикла и перейти к исполнению команды 4, если **false** – перейти к исполнению команды, следующей за командой цикла.
4. Увеличить значение счетчика цикла на 1.
5. Перейти к исполнению команды 2.

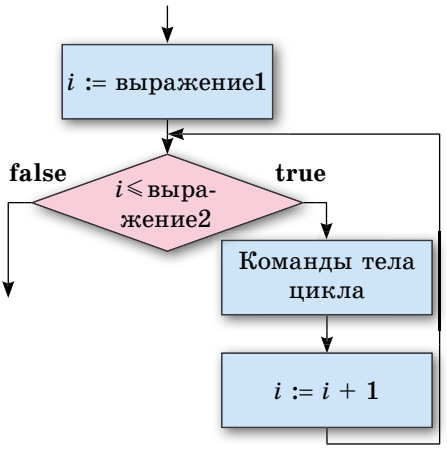


Рис. 6.30. Блок-схема команды цикла со счетчиком

Для иллюстрации исполнения команды цикла со счетчиком рассмотрим фрагмент проекта, в котором вычисляется сумма $1^2 + 2^2 + 3^2 + 4^2$.

```
var s, i: integer;
begin
  s := 0;
  for i := 1 to 4 do
  begin
    s := s + i*i;
  end;
  Label1.Caption := IntToStr (s);
end;
```

Исполним этот фрагмент программы.

Команда	*Результат выполнения*
s := 0	*s* = 0
i := 1	*i* = 1
i <= 4	(1 <= 4) = ***true***
s := *s* + *i***i*	*s* = 0 + 1*1 = 1
i := *i* + 1	*i* = 2
i <= 4	(2 <= 4) = ***true***
s := *s* + *i***i*	*s* = 1 + 2*2 = 5
i := *i* + 1	*i* = 3
i <= 4	(3 <= 4) = ***true***
s := *s* + *i***i*	*s* = 5 + 3*3 = 14
i := *i* + 1	*i* = 4
i <= 4	(4 <= 4) = ***true***
s := *s* + *i***i*	*s* = 14 + 4*4 = 30
i := *i* + 1	*i* = 5
i <= 4	(5 <= 4) = ***false***
Label1.Caption := *IntToStr* (*s*)	*Label1.Caption* = 30

Обращаем ваше внимание, что после окончания исполнения команды цикла со счетчиком значение счетчика цикла (переменной ***i***) на 1 больше значения **выражение2**, и это значение, при необходимости, можно использовать в следующих командах.

Для тех, кто хочет знать больше

Существует разновидность команды цикла со счетчиком, в котором счетчик не увеличивается на 1, а уменьшается на 1:

```
for <имя переменной> := <выражение1> downto <выражение2> do
begin
  <команды тела цикла>
end;
```

(англ. *down to* – вниз до).

Этот цикл продолжает исполняться, если значение счетчика цикла больше или равно (не меньше) значения **выражение2**.

Самое важное в этом пункте

Общий вид команды цикла со счетчиком:

```
for <имя переменной> := <выражение1> to <выражение2> do
begin
  <команды тела цикла>
end;
```

Строку **for <имя переменной> := <выражение1> to <выражение2> do** называют **строкой заголовка команды цикла со счетчиком**. Переменную в строке заголовка команды цикла со счетчиком, которая стоит перед знаком присваивания, называют **счетчиком цикла. Счетчик цикла, выражение1** и **выражение2** могут принимать только целые значения.

Дайте ответы на вопросы

1°. Что такое цикл в алгоритме?
2°. Как выглядит команда цикла со счетчиком в **Object Pascal**?
3•. Как исполняется команда цикла со счетчиком в **Object Pascal**?
4•. Как выглядит блок-схема цикла со счетчиком? Объясните исполнение такого фрагмента алгоритма.
5•. Могут ли команды тела цикла со счетчиком исполняться только один раз? Объясните свой ответ. Приведите пример.
6•. Могут ли команды тела цикла со счетчиком не исполняться ни разу? Объясните свой ответ. Приведите пример.
7*. Может ли исполнение цикла со счетчиком никогда не закончиться? Объясните свой ответ. Приведите пример.
8*. В чем заключаются различия в исполнении основных алгоритмических структур: следование, ветвление, цикл.

Выполните задания

1•. Составьте блок-схему фрагмента алгоритма и исполните его.

а)
```
n := 1;
for i := 1 to 4 do
n := n*i;
```

б)
```
p := 0; a := 2;
for i := 1 to 5 do
begin
  a := 3*a + 1;
  p := p + a;
end;
```

2•. Составьте блок-схему фрагмента алгоритма и исполните его.

а)
```
n := 0;
for i := 1 to 4 do
n := n + i*i*i;
```

б)
```
p := 0; a := 4;
for i := 1 to 5 do
begin
  a := 2*a − 1;
  p := p + a;
end;
```

3°. Составьте блок-схему алгоритма нахождения суммы пяти чисел, первое из которых равно 7, а каждое из четырех следующих на 3 больше предыдущего. Исполните алгоритм.

4°. Составьте блок-схему алгоритма нахождения суммы шести чисел, первое из которых равно 100, а каждое из пяти следующих в 5 раз меньше предыдущего. Исполните алгоритм.

5*. Составьте блок-схему алгоритма решения задачи: Определить, на сколько квадратов можно разрезать прямоугольную полоску бумаги с заданными длиной и шириной, если на каждом шагу от полоски отрезать квадрат, сторона которого равна меньшей из сторон полоски, оставшейся после предыдущего отрезания. Исполните алгоритм для двух различных значений исходных данных. Подберите эти значения так, чтобы команды тела цикла исполнились несколько раз; один раз.

6•. Создайте проект для вычисления суммы n чисел, первое из которых равно x, а каждое следующее на 5 больше предыдущего. Сохраните проект в папке **Задание 6.6.6**, созданной в вашей папке.

7•. Создайте проект для вычисления суммы n чисел, первое из которых равно x, а каждое следующее равно квадрату предыдущего. Сохраните проект в папке **Задание 6.6.7**, созданной в вашей папке.

8•. Создайте проект для решения задачи: Клиент положил в банк определенную сумму денег. Ежегодно банк добавляет ко вкладу 20 % от суммы, хранящейся на счету по результатам предыдущего года. Определить сумму, которая будет на счету через n лет. Сохраните проект в папке **Задание 6.6.8**, созданной в вашей папке.

9•. Создайте проект для решения задачи: Клиент положил в банк определенную сумму денег. Ежегодно банк добавляет ко вкладу заранее определенный процент от суммы, хранящейся на счету по результатам предыдущего года. Определить, на сколько увеличится сумма на счету через n лет. Сохраните проект в папке **Задание 6.6.9**, созданной в вашей папке.

6.7. ЦИКЛ С ПРЕДУСЛОВИЕМ

1. Какой вид имеет и как исполняется команда цикла со счетчиком в **Object Pascal**?
2. Какой вид имеет блок-схема цикла с предусловием и как исполняется этот цикл?
3. В каких случаях в алгоритмах используется цикл со счетчиком, а в каких – цикл с предусловием?

ЦИКЛ С ПРЕДУСЛОВИЕМ

Вы уже знаете, что **цикл с предусловием** как фрагмент алгоритма начинается с команды проверки условия, и результатом исполнения этой команды может быть либо **истина (Да, true)**, либо **ложь (Нет, false)**. И в зависимости от результата исполнения этой команды будут исполняться команды тела цикла, или команда алгоритма, следующая за циклом.

В общем случае в этой команде определяется значение некоторого логического выражения, которое может быть либо **true**, либо **false**.

Общий вид блок-схемы цикла с предусловием приведен на (рисунке 6.31).

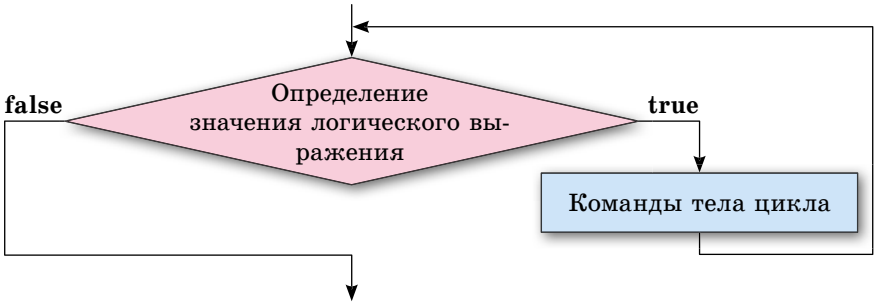


Рис. 6.31. Блок-схема цикла с предусловием

КОМАНДА ЦИКЛА С ПРЕДУСЛОВИЕМ В OBJECT PASCAL

Команда цикла с предусловием в языке программирования **Object Pascal** выглядит следующим образом:

```
While<логическое выражение>
Then begin
        <команды тела цикла>
     end;
```

(англ. *while* – пока).

Исполняется эта команда следующим образом: вычисляется значение логического выражения; если это значение **true**, то исполняются **команды тела цикла**, после чего опять вычисляется значение логического выражения, и если это значение опять **true**, то снова исполняются **команды тела цикла**, после чего опять вычисляется значение логического выражения; если значение логического выражения равно **false**, то **команды тела цикла** не исполняются, а исполняется команда, следующая за циклом.

Обращаем ваше внимание:

- если в теле цикла только одна команда, то операторные скобки **begin** и **end** можно не ставить;
- среди команд тела цикла могут быть и линейные фрагменты, и ветвления, и другие циклы.

Рассмотрим задачу, алгоритм решения которой содержит цикл с предусловием.

Задача. Сколько необходимо взять слагаемых, первое из которых равно заданному действительному числу, а каждое следующее на 3 больше предыдущего, чтобы их сумма превышала 100?

Данными этой задачи является первое слагаемое. Будем вводить его в поле.

Перед началом цикла присвоим переменной ***s***, которую используем для хранения суммы, значение первого слагаемого. А переменной ***n***, которую используем для хранения количества взятых слагаемых, присвоим значение 1, потому что в сумме уже учтено одно (первое) слагаемое.

К сумме следует прибавлять следующие слагаемые, пока эта сумма будет меньше или равна 100. Поэтому логическое выражение в заголовке цикла будет таким: ***s* <= 100**. В теле цикла необходимо искать следующее слагаемое, прибавлять его к текущему значению суммы и увеличивать количество слагаемых на 1.

Соответствующий фрагмент проекта будет выглядеть так:

```
var a, s: real; n: integer;
begin
  a := StrToFloat (Edit1.Text);
  s := a; n := 1;
  while s <=100 do
  begin
    a := a+3;
    s := s+a;
    n := n+1;
  end;
  Label1.Caption := IntToStr(n);
end;
```

Самое важное в этом пункте

Общий вид блок-схемы цикла с предусловием приведен на рисунке 6.31.

Команда цикла с предусловием в языке программирования **Object Pascal** выглядит так:

```
While <логическое выражение>
Then begin
        <команды тела цикла>
     end;
```

Исполненяется эта команда следующим образом: вычисляется значение логического выражения; если это значение **true**, то исполняются **команды тела цикла**, после чего снова вычисляется значение логического выражения, и если это значение опять **true**, то снова исполняются **команды тела цикла**, после чего опять вычисляется значение логического выражения; если значение логического выражения равно **false**, то **команды тела цикла** не исполняются, а исполняется команда, следующая за циклом.

Дайте ответы на вопросы

1•. Как выглядит блок-схема цикла с предусловием? Объясните исполнение этого цикла.

2•. Могут ли команды тела цикла с предусловием не исполняться ни разу? Объясните свой ответ. Приведите примеры.

3*. Может ли исполнение цикла с предусловием никогда не закончиться? Объясните свой ответ. Приведите примеры.

4*. Чем отличаются между собой цикл со счетчиком и цикл с предусловием?

5*. В чем различия в исполнении основных алгоритмических структур: следования, ветвления, цикл со счетчиком и цикл с предусловием?

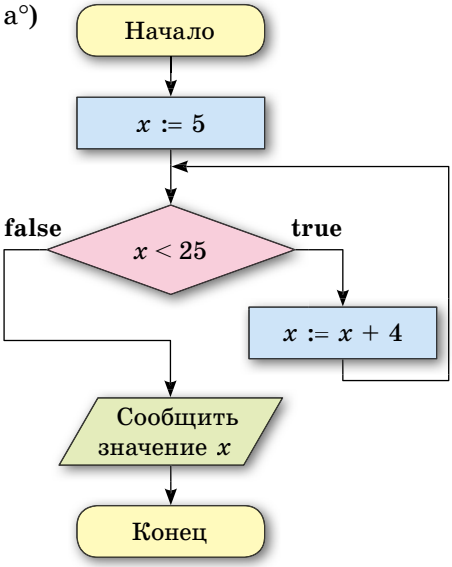
6°. Каков общий вид команды цикла с предусловием в **Object Pascal**?

7•. Как исполняется команда цикла с предусловием в **Object Pascal**?

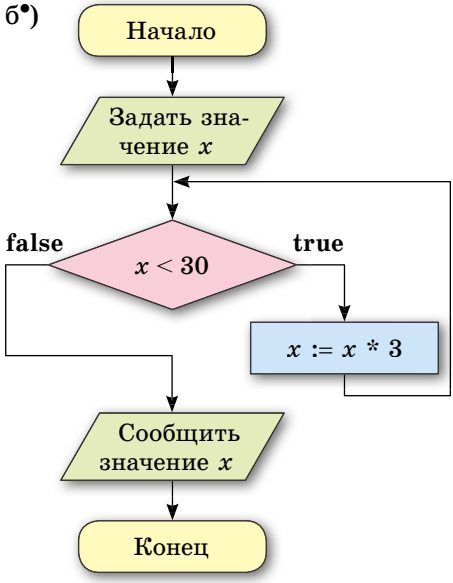
Выполните задания

1. Исполните алгоритмы.

а°)



б•)



2•. Исполните фрагмент программы и определите значения переменных после его завершения:

а)
```
k := 1; a := 12;
while a < 100 do
begin
  a := 2*a − 4;
  k := k + 1;
end;
```

б)
```
s := 0; a := 1; k := 1;
while a < 50 do
begin
  s := s + a;
  k := k + 1;
  a := k*k;
end;
```

3•. Исполните фрагмент программы и определите значения переменных после его завершения:

а)
```
k := 1; a := 100;
while a > 10 do
begin
  a := a/2;
  k := k + 1;
end;
```

б)
```
s := 0; a := 5;
while a <= 100 do
begin
  s := s + a;
  a := a*2;
end;
```

4•. Составьте блок-схему алгоритма нахождения суммы чисел, первое из которых равно 7, каждое следующее на 5 больше предыдущего и все они не превышают 100. Исполните алгоритм.

5•. Составьте блок-схему алгоритма нахождения суммы положительных чисел, первое из которых равно 50, каждое следующее на 8 меньше предыдущего. Исполните алгоритм.

6•. Создайте проект, в котором можно вычислить количество двузначных чисел среди чисел, первое из которых равно 10, а каждое следующее больше предыдущего на заданное положительное число. Сохраните проект в папке с именем **Задание 6.7.6**, созданной в вашей папке.

7•. Создайте проект для вычисления среднего арифметического положительных чисел, первое из которых равно 100, а каждое следующее меньше предыдущего на заданное положительное число. Сохраните проект в папке с именем **Задание 6.7.7**, созданной в вашей папке.

8•. Создайте проект для решения задачи: Клиент положил в банк определенную сумму денег. Ежегодно банк добавляет ко вкладу заранее определенный процент от суммы, хранящейся на счету по итогам предыдущего года. Через сколько лет сумма на счету превысит S грн? Сохраните проект в папке с именем **Задание 6.7.8**, созданной в вашей папке.

ПРАКТИЧЕСКАЯ РАБОТА № 12

«Составление и исполнение проектов с повторениями и ветвлениями для обработки величин»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте окно среды **Lazarus**.
2. Создайте проект для решения задачи: *Известны длины сторон двух прямоугольников. Вычислить площади прямоугольников и сравнить их. Определить, являются ли прямоугольники квадратами. Определить, поместится ли первый прямоугольник внутри второго.*
   1. Разместите на форме поля для ввода данных, метки для вывода результатов и три кнопки.
   2. В проекте:
      - создайте обработчик события **Click** для первой кнопки, исполнение которого приведет к вычислению площадей прямоугольников и их сравнению;
      - создайте обработчик события **Click** для второй кнопки, исполнение которого, в зависимости от установленных флажков, приведет к определению того, является ли соответствующий прямоугольник квадратом;

- составьте обработчик события **Click** для третьей кнопки, исполнение которого приведет к определению, поместится ли первый прямоугольник внутри второго или второй прямоугольник внутри первого, в зависимости от выбранного пункта раскрывающегося списка.

3. Сохраните проект в папке с именем **Практическая 12_1**, созданной в вашей папке.
4. Создайте проект для решения задачи: *Перед началом наводнения уровень воды в реке составлял **H** метров. Во время наводнения каждый час уровень воды увеличивался на **P** процентов по сравнению с уровнем предыдущего часа. Каким будет уровень воды через **N** часов после начала наводнения? Через сколько часов после начала наводнения уровень воды будет не менее чем **K** метров?*
   1. Разместите на форме поля для ввода данных, метки для вывода результатов и три кнопки.
   2. Установите на первой кнопке надпись **Задание 1**, на второй – **Задание 2**, на третьей – **Сначала**, у полей – *пустой текст*.
   3. Создайте обработчик события **Click** для первой кнопки, исполнение которого приведет к выводу в окно сообщения ответа на первый вопрос задачи.
   4. Исполните проект и убедитесь, что результаты его работы правильные.
   5. Создайте обработчик события **Click** для второй кнопки, исполнение которого приведет к получению ответа на второй вопрос задачи и выводу его в отдельную метку.
   6. Исполните проект и убедитесь, что результаты его работы правильные.
   7. Создайте обработчик события **Click** для третьей кнопки, исполнение которого приведет к очистке текста в полях и метке с ответом на второй вопрос задачи.
5. Сохраните проект в папке с именем **Практическая 12_2**, созданной в вашей папке.
6. Закройте окно среды **Lazarus**.

6.8. ОТОБРАЖЕНИЕ ГРАФИЧЕСКИХ ОБЪЕКТОВ СРЕДСТВАМИ ЯЗЫКА ПРОГРАММИРОВАНИЯ

1. Какие графические примитивы можно создавать в графическом редакторе и текстовом процессоре?
2. Какие свойства графических примитивов вам известны?
3. Какие общие свойства метки, кнопки и формы вы знаете?

СРЕДСТВА ДЛЯ РИСОВАНИЯ В OBJECT PASCAL

Графические объекты в проектах, разработанных в **Lazarus**, можно создавать и отображать разными способами. Одним из них является рисование графических примитивов по их координатам на координатной плоскости.

Часть такой координатной плоскости размещена, например, на форме. Начало координат совпадает с верхним левым углом формы. Ось ***Ox*** направлена вправо, ось ***Oy*** – вниз. Часть координатной плоскости, которая отображается на форме, имеет размеры ширины и высоты формы. На рисунке 6.32 красным цветом обозначена точка с координатами (100, 200).

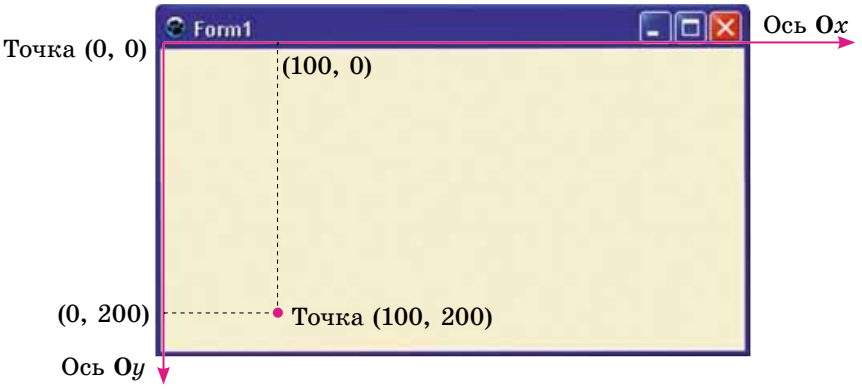


Рис. 6.32. Часть координатной плоскости на форме

Для рисования на форме используется ее свойство **Canvas** (англ. *canvas* – полотно, канва).

Каждая точка (пиксель) на форме имеет свои координаты и цвет. Координаты могут быть заданы только целыми числами. Цвет можно задать константой типа **TColor**. Примеры констант и соответствующих им цветов приведены в таблице 6.1.

Таблица 6.1

Константы для цветов в Lazarus

Константа	*Цвет*	*Константа*	*Цвет*
clBlack	Черный	*clSilve*	Серебристый
clWhite	Белый	*clGray*	Серый
clRed	Красный	*clGreen*	Зеленый
clBlue	Синий	*clNavy*	Темно-синий
clLime	Салатовый	*clOlive*	Оливковый

Для того чтобы установить для точки с координатами (*x*, *y*) некоторый цвет, например красный, необходимо исполнить команду:

```
Canvas.Pixels[x, y] := clRed;
```

Для создания графических примитивов можно использовать команды, приведенные в таблице 6.2.

Таблица 6.2

Команды для рисования графических примитивов на координатной плоскости формы

Команда	*Объяснение*
Canvas.MoveTo (*x*, *y*)	Выбор на форме точки (*x*, *y*), с которой будет начинаться рисование линий
Canvas.LineTo (*x*, *y*)	Рисование отрезка от последней точки, в которой происходило рисование, до точки (*x*, *y*). Последовательное использование нескольких таких команд приведет к рисованию отрезка или ломаной
Canvas.Rectangle (*x*1, *y*1, *x*2, *y*2)	Рисование прямоугольника, у которого противоположные вершины расположены в точках (*x*1, *y*1) и (*x*2, *y*2), а стороны параллельны осям координат
Canvas.Ellipse (*x*1, *y*1, *x*2, *y*2)	Рисование эллипса, вписанного в прямоугольник, у которого противоположные вершины расположены в точках (*x*1, *y*1) и (*x*2, *y*2), а стороны параллельны осям координат. Прямоугольник при этом не отображается
Canvas.Pie (*x*1, *y*1, *x*2, *y*2, *x*3, *y*3, *x*4, *y*4)	Рисование замкнутой фигуры – сектора эллипса. Точки (*x*1, *y*1) и (*x*2, *y*2) задают прямоугольник, в который вписан эллипс. Начальная точка дуги определяется пересечением эллипса с прямой, проходящей через его центр и точку (*x*3, *y*3). Конечная точка дуги определяется пересечением эллипса с прямой, проходящей через его центр и точку (*x*4, *y*4). Дуга рисуется против часовой стрелки от начальной до конечной точки (рис. 6.33).

В приведенном на рисунке 6.33 примере эллипс вписан в прямоугольник, имеющий противоположные вершины в точках *A* (100, 100) и *C* (500, 300). Сектор эллипса, образован отрезками, соединяющими центр прямоугольника (точку *T* (300, 200)) с точками *M* (300, 100) и *N* (400, 100), и

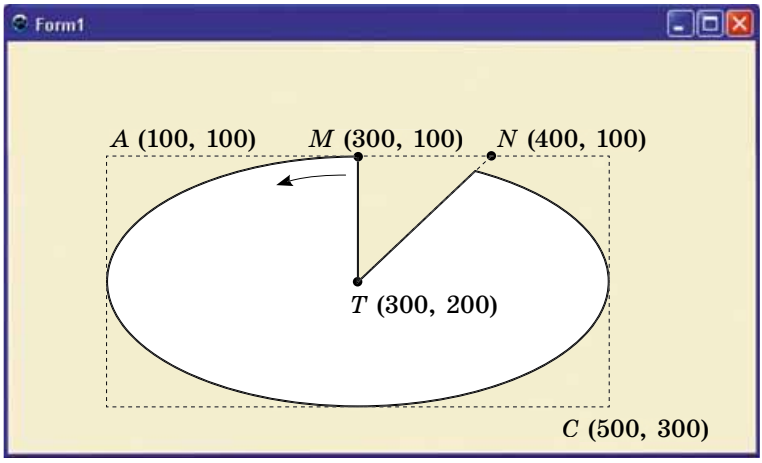


Рис. 6.33. Сектор эллипса, созданный командой **Canvas.Pie** (100, 100, 500, 300, 300, 100, 400, 100)

дугой эллипса, начинающейся в точке *M* (300, 100), а заканчивающейся при пересечении эллипса с отрезком *TN*.

Для каждого примитива можно задать значение свойств его контура и внутренней области. Значение свойств контура и внутренней области графического примитива совпадают со значениями свойств формы, приведенными в таблице 6.3.

Таблица 6.3

Свойства формы для установки значений свойств контура и внутренней области графических примитивов

Свойство	*Назначение и примеры значений*	*Значение по умолчанию*
Canvas.Pen.Color	Цвет линии контура	Черный
Canvas.Pen.Width	Толщина линий контура	1 пиксель
Canvas.Pen.Style	Стиль линии контура, например, *psSolid* – сплошная линия, *psDash* – штриховая линия, *psDot* – пунктирная линия и др.	Сплошная
Canvas.Brush.Color	Цвет заливки внутренней области	Белый
Canvas.Brush.Style	Стиль заливки внутренней области, например, *bsSolid* – сплошная заливка, *bsCross* – заливка в виде сетки с горизонтальными и вертикальными линиями, *bsDiagCross* – заливка в виде сетки с диагональными линиями и др.	Сплошная

Создавая изображение, состоящее из графических примитивов, следует сначала установить значения свойств контура и внутренней области фигуры, после чего отобразить саму фигуру. На рисунке 6.34 показан результат исполнения процедуры создания цветного прямоугольника:

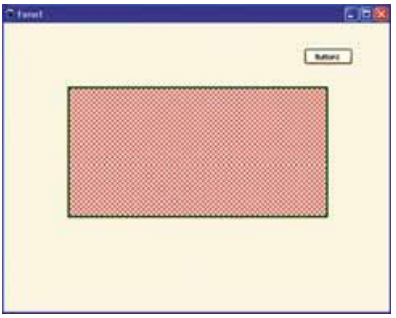


Рис. 6.34. Результат исполнения процедуры рисования графического примитива

```
procedure Tform1.Button1Click(Sender: Tobject);
begin
  Canvas.Pen.Color := clGreen; // цвет контура зеленый
  Canvas.Pen.Width := 3;  // толщина контура 3 пикселя
  Canvas.Brush.Color := clRed; // цвет заливки красный
  Canvas.Brush.Style := bsDiagCross; (* стиль заливки – сетка с диагональными линиями *)
  Canvas.Rectangle (100, 100, 500, 300); (* прямоугольник с противоположными вершинами в точках (100, 100) и (500, 300) и сторонами, параллельными осям координат *)
end;
```

Для того чтобы рисунок отображался на форме сразу после запуска проекта на исполнение, необходимо команды рисования размещать в процедуре – обработчике события **Paint** для формы.

КОМПОНЕНТ ФИГУРА ДЛЯ СОЗДАНИЯ ГРАФИЧЕСКИХ ПРИМИТИВОВ

Графические примитивы в проектах **Lazarus** можно создавать, используя компонент **Фигура**.

Чтобы разместить **Фигуру** на форме, следует дважды щелкнуть на кнопке **TShape** (англ. *shape* – форма, фигура) на вкладке **Additional** (англ. *additional* – дополнительный) окна среды **Lazarus**.

Свойства **Фигуры Name**, **Enabled**, **Visible** аналогичные соответствующим свойствам формы. Как и для формы, значения свойств **Фигуры Top**, **Left**, **Height** и **Width** можно изменить или на вкладке **Свойства**, или перетаскиванием.

Компонент **Фигура** имеет свойства, определяющие цвет и стиль контура и внутренней области (табл. 6.4).

Таблица 6.4

Свойства компонента Фигура

Свойство	*Что определяет ее значение*
Pen.Color	Цвет контура
Pen.Width	Толщина контура
Pen.Style	Стиль линии контура
Brush.Color	Цвет заливки внутренней области
Brush.Style	Стиль заливки внутренней области

Значение этих свойств можно устанавливать на вкладке **Свойства** в окне **Инспектор объектов** или с использованием команды присваивания. Эти значения могут быть такими же, как и для графических примитивов.

Кроме того, **Фигура** имеет свойство **Shape**, определяющее ее форму. Это свойство может принимать, например, значения: *stRectangle* – прямоугольник, *stCircle* – круг, *stSquare* – квадрат, *stDiamond* – ромб, *stTriangle* – треугольник и др. По умолчанию значение этого свойства определяет квадрат с контуром черного цвета толщиной 1 пиксель и сплошной заливкой белого цвета.

На рисунке 6.35 приведен пример изображения, созданного с использованием нескольких объектов **Фигура** с различными значениями свойств.

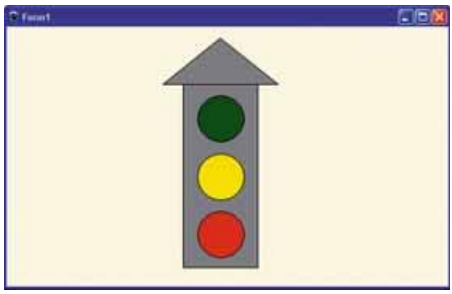


Рис. 6.35. Создание изображения с использованием объектов **Фигура**

ОТОБРАЖЕНИЕ ИЗОБРАЖЕНИЙ ИЗ ВНЕШНИХ ФАЙЛОВ

В проектах можно отображать изображения, хранящиеся во внешних файлах. Для этого предназначен компонент **Область изображения**. Для его размещения на форме следует дважды щелкнуть на кнопке **TImage** (англ. *image* – изображение) на вкладке **Additional** в окне среды **Lazarus**.

Свойства **Области изображения Name**, **Enabled**, **Visible** аналогичны соответствующим свойствам формы. Как и для формы, значения свойств **Области изображения Top**, **Left**, **Height** и **Width** можно изменить или на вкладке **Свойства**, или перетаскиванием.

Отображение изображения, хранящегося во внешнем файле, определяется значениями таких свойств (табл. 6.5):

Таблица 6.5

Свойства компонента Область изображения

Свойство	*Объяснение*
Picture	Выбор файла с графическим изображением для отображения в **Области изображения**
AutoSize	При значении **true** компонент **Область изображения** принимает размеры изображения, полученного из файла. При значении **false** размеры компонента не зависят от размера изображения, а определяются значениями соответствующих свойств
Stretch	При значении **true** изображение, полученное из файла, принимает размеры **Области изображения**. При значении **false** изображение сохраняет свои размеры и не зависит от размеров компонента
Proportional	При значении **true** ширина и высота изображения, полученного из файла, будут изменяться пропорционально при размещении в **Области изображения**. При значении **false** ширина и высота изображения могут изменяться не пропорционально

Пример отображения в проекте изображения, вставленного из внешнего файла, приведен на рисунке 6.36.



Рис. 6.36. Отображение в проекте изображения из внешнего файла

Самое важное в этом пункте

Графические примитивы можно рисовать по их координатам на координатной плоскости, часть которой размещена на форме. Для этого используется ее свойство **Canvas**. Каждая точка на форме имеет свои координаты и цвет. Для каждого графического примитива можно задать значения свойств его контура и внутренней области.

Графические примитивы в проектах **Lazarus** можно создавать также, использовав компонент **Фигура**. Этот компонент имеет свойства, значения которых определяют цвет и стиль контура и внутренней области.

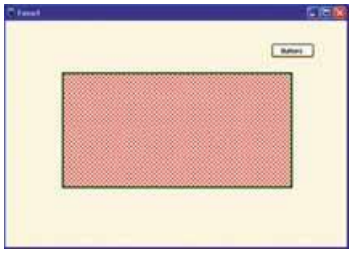
Можно отображать изображения, содержащиеся во внешних файлах. Для этого предназначен компонент **Область изображения**.

Дайте ответы на вопросы

1•. Как размещена на форме координатная плоскость?
2°. Какое свойство формы используется для рисования графических примитивов?
3•. Какие команды используются для рисования графических примитивов – линия, ломаная, прямоугольник, эллипс, сектор?
4•. Как изменить контур и внутреннюю область графических примитивов?
5°. Какой компонент используется для создания графических примитивов? Как его установить на форме?
6•. Значение каких свойств **Фигуры** определяют вид ее контура и внутренней области?
7•. Значение какого свойства **Фигуры** определяет ее форму? Какие значения оно может принимать?
8°. Какой компонент используется для отображения на формах графических изображений, содержащихся во внешних файлах? Как установить его на форме?
9•. Как выбрать файл для отображения в **Области изображения**?

Выполните задания

1°. Создайте проект, при исполнении которого после выбора кнопки будет нарисован на форме прямоугольник по образцу, приведенному на рисунке. Сохраните проект в папке с именем **Задание 6.8.1**, созданной в вашей папке.



К заданию 1

2•. Создайте проект, после запуска которого на форме будет нарисовано изображение следующих графических примитивов:

а) отрезок от точки (10, 10) до точки (200, 100) красного цвета, тип линии – штрих-пунктир, толщина линии – 5;

б) ломаная из четырех отрезков синего цвета, стиль линии – точками, толщина – 3;

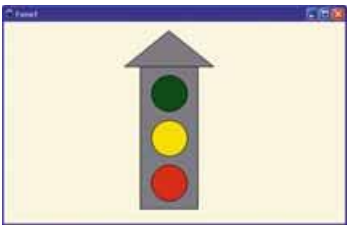
в) прямоугольник с противоположными вершинами в точках (50, 50) и (250, 150), линии контура – красные сплошные толщиной 1, заливка серебристого цвета, стиль заливки – горизонтальные линии;

г) эллипс, вписанный в прямоугольник с вершинами в точках (100, 20) и (200, 300), линии контура – желтые пунктирные толщиной 2, заливка салатового цвета, стиль заливки – вертикальные линии;

д) сектор эллипса, вписанного в прямоугольник с вершинами (50, 50) и (150, 150), начальная точка дуги (150, 100), конечная, – (100, 150), линии контура – белые сплошные толщиной 4, заливка оливкового цвета, стиль заливки – наклонные линии.

Сохраните проект в папке с именем **Задание 6.8.2**, созданной в вашей папке.

3•. Создайте проект, при исполнении которого будет построено изображение из фигур в соответствии с образцом на рисунке. Сохраните проект в папке с именем **Задание 6.8.3**, созданной в вашей папке.



К заданию 3

4•. Создайте проект, при исполнении которого из фигур будет построено изображение:
а) грузовика;
б) робота;
в) компьютера;
г) ракеты.

Стиль и цвет фигур выберите самостоятельно. Сохраните проект в папке с именем **Задание 6.8.4**, созданной в вашей папке.

5°. Создайте проект, при исполнении которого на форме будет отображаться изображение, содержащееся в файле, например **Глава 6\Пункт 6.8\задача 6.8.5.jpg**. Сохраните проект в папке с именем **Задание 6.8.5**, созданной в вашей папке.

6•. Создайте проект, при исполнении которого на форме в качестве фона будет размещено изображение из файла, например **Глава 6\Пункт 6.8\задания 6.8.6.jpg**. Поверх него из фигур создайте изображение автомобиля. Сохраните проект в папке с именем **Задание 6.8.6**, созданной в вашей папке.

7*. Разместите на форме кнопку, фигуру, три флажка и поле с раскрывающимся списком. Создайте проект, в котором при выборе кнопки в зависимости от выбранного значения в списке **Фигура** будет менять форму, принимая значения квадрата, круга или треугольника, а в зависимости от выбранных флажков контур будет иметь толщину 1 или 4, цвет заливки будет желтым или зеленым, стиль заливки – сплошной или диагональные линии. Сохраните проект в папке с именем **Задание 6.8.7**, созданной в вашей папке.

ПРАКТИЧЕСКАЯ РАБОТА № 13

«Составление и исполнение алгоритмов с графическим отображением данных»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте окно среды **Lazarus**.
2. Создайте проект, после запуска которого на исполнение:
   а) в верхней части формы из графических примитивов будет изображено солнце;
   б) в нижней части формы будет размещено изображение из файла **Глава 6\Пункт 6.8\практическая 13.jpg**;
   в) из фигур будет создано изображение дома.
3. Сохраните проект в папке с именем **Практическая 13**, созданной в вашей папке.
4. Запустите проект на исполнение.
5. Проанализируйте полученный результат.
6. Закройте окно исполнения проекта.
7. Закройте окно среды **Lazarus**.

Глава 7. Обработка числовых данных

В этой главе вы узнаете:

- об абсолютных, относительных и смешанных ссылках на ячейки и диапазоны ячеек
- о создании и настройке диаграмм различного типа, выборе типа диаграммы
- об установке параметров страницы и печати электронной таблицы
- об условном форматировании
- о назначении и использовании математических, статистических, логических функций табличного процессора
- об упорядочении данных в таблицах
- об автоматических и расширенных фильтрах, промежуточных итогах



7.1. АБСОЛЮТНЫЕ, ОТНОСИТЕЛЬНЫЕ И СМЕШАННЫЕ ССЫЛКИ. УСТАНОВКА ПАРАМЕТРОВ СТРАНИЦЫ. ПЕЧАТЬ ЭЛЕКТРОННОЙ ТАБЛИЦЫ

1. Как в табличном процессоре записываются ссылки на ячейки? Для чего их используют?
2. Что такое модификация формул? Как она происходит? По каким правилам осуществляется?
3. Какие свойства имеет страница текстового документа? Какие значения эти свойства могут приобретать? Как их установить?

ОТНОСИТЕЛЬНЫЕ, АБСОЛЮТНЫЕ И СМЕШАННЫЕ ССЫЛКИ

В седьмом классе вы уже изучали, что в электронных таблицах в формулах можно использовать ссылку на ячейки или диапазоны ячеек. Они обеспечивают автоматический пересчет значений по формулам, в которых используются ссылки. Также вы знаете, что при копировании формул ссылки в них модифицируются. Однако иногда необходимо, чтобы при копировании формул некоторые ссылки не модифицировались.

Ссылки, которые модифицируются при копировании формулы, называют *относительными*.

Ссылки, которые не модифицируются при копировании формулы, называют *абсолютными*.

Для того чтобы ссылка была абсолютной, следует перед номером столбца и перед номером строки ввести символ $.

Например, ссылка **B10** является относительной, а ссылка **$B$10** – абсолютной.

Рассмотрим пример, который объясняет целесообразность использования абсолютных ссылок.

Пусть в ячейках диапазона **C3:C7** (рис. 7.1) содержится число отличников в каждом 8–10-ом классах школы, а в ячейке **C8** – их общее количество. В ячейках диапазона **D3:D7** необходимо вычислить процент отличников в каждом из этих классов по отношению к общему количеству отличников.

D3 =C3/$C$8*100

	A	B	C	D
1				
2			Отличники	Процент
3		8 А	3	18,75
4		8 Б	1	6,25
5		9 А	5	31,25
6		9 Б	4	25,00
7		10	3	18,75
8		Всего	16	
9				

Рис. 7.1. Использование относительных и абсолютных ссылок

Для вычисления процентов в ячейках диапазона **D3:D7** должны находиться следующие формулы:

D3 = C3/C8*100
D4 = C4/C8*100
D5 = C5/C8*100
D6 = C6/C8*100
D7 = C7/C8*100

Если проанализировать эти формулы, то можно сделать вывод, что ссылки в числителе формулы меняются, а

ссылки в знаменателе – нет. Если мы введем в ячейку **D3** формулу **=C3/C8*100** и скопируем ее в ячейки диапазона **D4:D7**, то все ссылки в формуле будут модифицироваться. Однако необходимо сделать так, чтобы ссылка **C3** при копировании формулы модифицировалась, а ссылка **C8** – нет. То есть формула в ячейке **D3** должна иметь следующий вид: **=C3/$C$8*100**.

Если в ссылке символ $ добавлен только перед номером столбца или только перед номером строки, например **$B10** или **B$10**, то при копировании такие ссылки модифицируются частично: не меняется только номер столбца или только номер строки, рядом с которым стоит символ $. Такие ссылки называют **смешанными**.

Ссылка, в которой при копировании формулы модифицируется или номер столбца, или номер строки, называют *смешанными*.

Изменить вид ссылки в формуле можно во время ее ввода или при редактировании формулы последовательным нажатием клавиши **F4**. При этом виды ссылки меняются по очереди в следующем порядке: относительная – абсолютная – смешанная.

Если в формуле для ссылки на ячейку использовать не адрес, а ее имя, то при копировании этой формулы такая ссылка модифицироваться не будет. То есть, **имя ячейки в формуле является абсолютной ссылкой на эту ячейку**. Поэтому в рассмотренном выше примере можно ячейке **C8** дать имя, например **Отличники**, и тогда формула в ячейке **D3** будет выглядеть так: **=C3/Отличники*100**.

Примеры абсолютных, относительных и смешанных ссылок представлены в таблице 7.1.

Таблица 7.1

Типы ссылок

Ссылка	*Тип*	*Объяснение*
C3	Относительная	При копировании формулы может изменяться и номер столбца, и номер строки
$A$1	Абсолютная	При копировании формулы номер столбца и номер строки неизменны
курс	Абсолютная	При копировании формулы номер столбца и номер строки неизменны
C$1	Смешанная	При копировании формулы номер столбца может изменяться, а номер строки меняться не будет
$A3	Смешанная	При копировании формулы номер столбца будет неизменным, а номер строки может изменяться

УСТАНОВКА ЗНАЧЕНИЙ СВОЙСТВ СТРАНИЦЫ

Вы уже умеете печатать текстовые документы в **Word 2007** и знаете, что предварительно следует установить значения свойств страницы документа (размер, ориентацию, поля и т. д.), затем в режиме **Предварительного просмотра** просмотреть внешний вид документа и потом печатать его.

Аналогично происходит печать и электронной таблицы. Однако имеются и определенные отличия выполнения этих операций в **Excel 2007**.

Установка значений свойств страницы электронной таблицы осуществляется элементами управления группы **Параметры страницы** на вкладке **Разметка страницы** или на вкладках диалогового окна **Параметры страницы** (рис. 7.2).

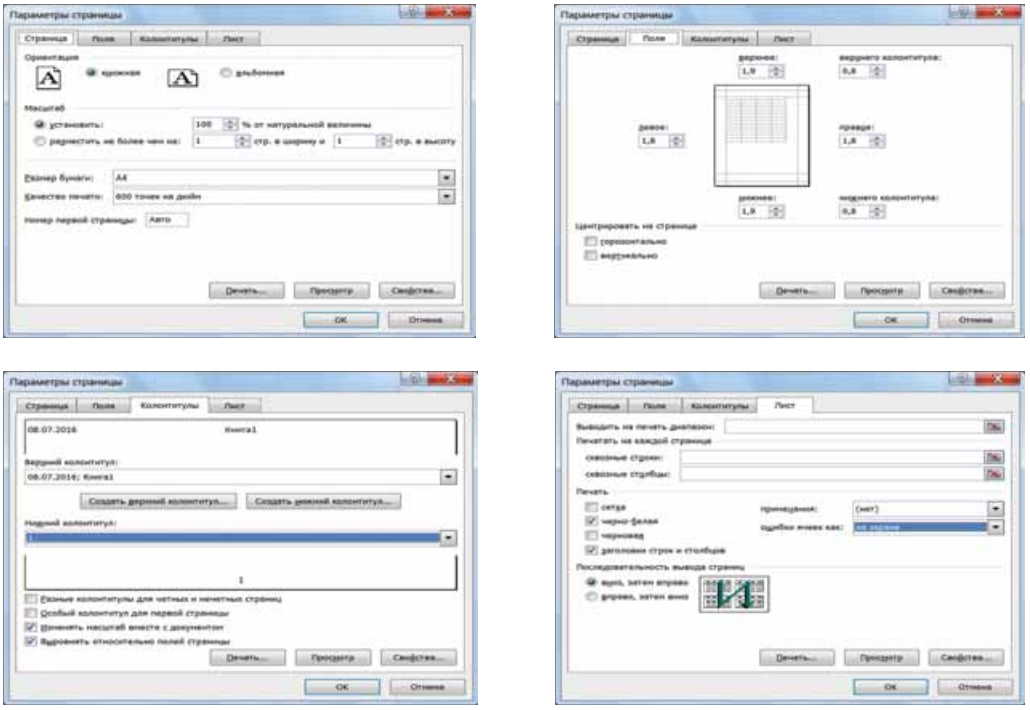


Рис. 7.2. Вкладки диалогового окна **Параметры страницы**

На вкладке **Страница** можно установить значение следующих свойств:

- ориентация страницы, на которой будет осуществляться печать;
- масштаб в процентах для той части электронной таблицы, которая будет напечатана;
- размер листа бумаги для печати;
- качество печати;
- номер первой страницы

и другое.

На вкладке **Поля** можно установить такие значения: размеры полей на странице, размеры колонтитулов, центрирование таблицы на странице.

На вкладке **Колонтитулы** можно создать верхний и нижний колонтитулы, выбрав соответствующую кнопку или нужный вид колонтитула в списках **Верхний колонтитул** / **Нижний колонтитул**.

На вкладке **Лист** можно установить значение таких свойств:

- область печати – диапазон ячеек листа электронной таблицы, которые будут выводиться на печать;

- печать сквозных строк и столбцов на каждой странице – диапазон ячеек, которые будут повторяться на каждой странице, если напечатанная часть таблицы на бумаге занимает больше одной страницы;
- порядок печати страниц – вниз, затем вправо или влево, затем вниз и другое.

ПЕЧАТЬ ЭЛЕКТРОННОЙ ТАБЛИЦЫ

Для того чтобы увидеть, как будет выглядеть электронная таблица на бумаге, следует перейти в режим **Предварительный просмотр** (рис. 7.3), который устанавливается ***Office*** ⇒ ***Печать*** ⇒ ***Предварительный просмотр***, или в режиме просмотра электронной таблицы **Разметка страницы** (рис. 7.4), который устанавливается выбором кнопки в **Строке состояния**.

В режиме **Предварительный просмотр** будет отображаться подготовленная к печати таблица, которая разделена на страницы. Если область печати не умещается на одной странице листа бумаги, то **Excel 2007** автоматически разбивает эту область на несколько страниц (такой случай показан на рис. 7.4).



Рис. 7.3. Режим **Предварительный просмотр**

	A	B	C	D	E	F	G
1	Код	Магазин	Жанр	Название фильма	Цена	Количество	Продано
2	546	Мир вокруг	Боевик	Терминатор	9	2	18
3	621	Сатурн	Фантастика	Звездные войны	13	3	39
4	876	Континент	Драма	Жестяной кубок	7	3	21
5	999	Континент	Фильм ужасов	Закат-убежище вампиров	4	3	12
6	432	Сатурн	Фантастика	Назад в прошлое	12	4	48
7	677	Стиль	Фантастика	Пангольеры	14	4	56
8	888	Континент	Фильм ужасов	Иствикские ведьмы	7	4	28
9	999	Мир вокруг	Фильм ужасов	Закат-убежище вампиров	4	5	20
10	543	Мир вокруг	Драма	Под прессом	8	6	48
11	654	Стиль	Драма	Уолл-стрит	7	6	42
12	876	Сатурн	Драма	Жестяной кубок	7	6	42
13	987	Стиль	Фильм ужасов	Армия тьмы	5	6	30
14	432	Мир вокруг	Фантастика	Назад в прошлое	12	7	84
15	543	Мир вокруг	Комедия	Близнецы	6	7	42
16	543	Континент	Комедия	Близнецы	6	7	42
17	543	Континент	Драма	Под прессом	8	7	56
18	546	Стиль	Боевик	Терминатор	9	7	63
19	654	Континент	Драма	Уолл-стрит	7	7	49
20	678	Сатурн	Комедия	Бум 1,2	6	7	42
21	765	Мир вокруг	Драма	Аптечный ковбой	5	7	35
22	987	Мир вокруг	Фильм ужасов	Армия тьмы	5	7	35
23	543	Сатурн	Драма	Под прессом	8	8	64
24	654	Сатурн	Драма	Уолл-стрит	7	8	56
25	765	Сатурн	Драма	Аптечный ковбой	5	8	40
26	888	Сатурн	Фильм ужасов	Иствикские ведьмы	7	8	56
27	987	Сатурн	Фильм ужасов	Армия тьмы	5	8	40
28	999	Сатурн	Фильм ужасов	Закат-убежище вампиров	4	8	32
29	123	Стиль	Боевик	Крепкий орешек	13	9	117
30	543	Сатурн	Комедия	Близнецы	6	9	54
31	543	Стиль	Драма	Под прессом	8	9	72
32	765	Континент	Драма	Аптечный ковбой	5	9	45
33	876	Стиль	Драма	Жестяной кубок	7	9	63
34	888	Стиль	Фильм ужасов	Иствикские ведьмы	7	9	63
35	999	Стиль	Фильм ужасов	Закат-убежище вампиров	4	9	36

Рис. 7.4. Режим просмотра **Разметка страницы**



Рис. 7.5. Окно **Печать**

В режиме **Предварительный просмотр** можно просматривать страницы по одной, а в режиме **Разметка страницы** – видеть сразу несколько страниц, разделенных штриховыми линиями. Перемещая штриховые линии, можно изменять распределение содержимого по страницам. После закрытия режима **Предварительный просмотр** на листе таблицы появляются вертикальные и горизонтальные штриховые линии, которые демонстрируют разбитие таблицы на страницы.

Если в окне прграммы на панели быстрого доступа имеется кнопка **Быстрая печать**  , то ее выбор печатает одну копию части текущего листа книги, которая заполнена данными, со значениями свойств печати по умолчанию. Это же можно осуществить, выполнив ***Office*** ⇒ ***Печать*** ⇒ ***Быстрая печать***.

Если необходимо установить другие параметры печати, то следует выполнить ***Office*** ⇒ ***Печать*** ⇒ ***Печать*** и в открывшемся окне **Печать** (рис. 7.5) установить нужные значения: доступный принтер; количество копий, которые будут напечатаны; объект для печати (выделенный диапазон, выделенные листы или всю книгу) и т. п. С помощью переключателей **все** и **страницы** можно указать номера отдельных страниц или их диапазонов для печати.

Самое важное в этом пункте

В формулах можно использовать различные виды ссылок на ячейки: **относительные** (модифицируются при копировании формулы), **абсолютные** (не модифицируются при копировании формулы), **смешанные** (модифицируются частично).

Для того чтобы ссылка была абсолютной, следует перед номером столбца и номером строки ввести символ $. В смешанных ссылках символ $ вводится или возле номера столбца, или возле номера строки.

Если в формуле для ссылки на ячейку использовать ее имя, то при копировании этой формулы такая ссылка модифицироваться не будет, она считается абсолютной.

Установка значений параметров страницы электронной таблицы для печати осуществляется на **Ленте** элементами управления группы **Параметры страницы** на вкладке **Разметка страницы** или на вкладках диалогового окна **Параметры страницы** этой группы.

Для того чтобы просмотреть, как будет выглядеть электронная таблица на бумаге, следует перейти в режим **Предварительный просмотр**, который устанавливается ***Office*** ⇒ ***Печать*** ⇒ ***Предварительный просмотр***, или в режиме просмотра электронной таблицы **Разметка страницы**, который устанавливается выбором одноименной кнопки в **Строке состояния**.

Для того чтобы напечатать электронную таблицу, следует выполнить ***Office*** ⇒ ***Печать*** ⇒ ***Быстрая печать*** (со значениями свойств страницы по умолчанию) или ***Office*** ⇒ ***Печать*** ⇒ ***Печать***.

Дайте ответы на вопросы

1°. Какие виды ссылок на ячейки можно использовать в формулах в электронных таблицах?
2°. Как записать относительную, абсолютную и смешанную ссылки на ячейку **B2**?
3•. Чем отличается использование различных видов ссылок?
4°. К какому виду ссылок относится имя ячейки?
5°. Какие свойства страницы электронной таблицы вы знаете?
6•. Как установить значения свойств печати электронной таблицы?
7•. Для чего используют предварительный просмотр электронной таблицы? Каковы его возможности?
8•. Как просмотреть, на какие страницы будет разделена электронная таблица при печати?
9°. Как выполнить быструю печать электронной таблицы со значениями свойств по умолчанию?
10•. Как напечатать электронную таблицу, установив новые параметры печати?

Выполните задание

1•. В ячейке **E3** находится формула: **=$B$3+C4**. Запишите, как будет выглядеть эта формула, если ее скопировать в ячейку:

а) **E7;** в) **C3;** д) **K15;** ж) **C3;**
б) **E7;** г) **E11;** е) **T34;** з) **A2.**

2•. В ячейке **K8** находится формула: **=E5+$C3**. Запишите, как будет выглядеть эта формула, если ее скопировать в ячейку:

а) **C9;** в) **B4;** д) **D8;** ж) **I2;**
б) **K7;** г) **E9;** е) **T23;** з) **F5.**

3•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.1\образец 7.1.3.xlsx**).

а•) На листе **Лист1** в ячейку **F3** запишите формулу для нахождения прибыли фирмы за год, которую затем скопируйте с использованием маркера заполнения в ячейки **F4:F9**. Какие ссылки должна содержать эта формула: абсолютные, относительные или смешанные? Ответ объясните.

б•) На листе **Лист2** в ячейку **E4** запишите формулу для определения цены товара в гривнях (курс евро находится в ячейке **B3**). Скопируйте формулу с использованием маркера заполнения в ячейки **E5:E12**. Какие ссылки должна содержать эта формула: абсолютные, относительные или смешанные? Ответ объясните.

в*) На листе **Лист3** в ячейку **D10** введите формулу для вычисления общего количества учащихся в 8-х классах. Дайте этой ячейке имя **Ученики_8**. Используйте это имя в формулах для вычисления процентов количества учащихся 8-А класса по отношению к общему количеству учащихся 8-х классов. Скопируйте эту формулу с использованием маркера заполнения в ячейки для нахождения процентов количества учащихся других 8-х классов по от-

ношению к общему количеству учащихся 8-х классов. Состоялась ли модификация формулы? Ответ объясните.

Сохраните книгу в вашей папке в файле с именем **задание 7.1.3.xlsx**.

4*. Создайте таблицу для определения стоимости украинских банкнот в долларах, евро и фунтах стерлингов. Предусмотрите сохранение курсов доллара, евро и фунтов стерлингов в отдельных ячейках. Сохраните книгу в вашей папке в файле с именем **задание 7.1.4.xlsx**.

5•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.1\образец 7.1.5.xlsx**). Выполните предварительный просмотр страниц электронной таблицы на листе **Лист1** двумя способами. Выполните печать этого листа электронной таблицы.

ПРАКТИЧЕСКАЯ РАБОТА № 14

«Решение задачи на вычисления»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте указанный учителем файл (например, **Глава 7\Практическая 14\образец.xlsx**).
2. На листе **Лист1** в ячейку **F2** введите формулу для нахождения пути, который прошел первый турист за три дня. Скопируйте формулу с использованием маркера заполнения в ячейки **F3:F7**.
3. На листе **Лист2** в ячейку **E3** введите формулу для определения цены товара в долларах (курс доллара находится в ячейке **B2**), которую затем скопируйте с использованием маркера заполнения в ячейки **E4:E10**.
4. На листе **Лист3** в ячейку **B10** введите формулу для вычисления общего количества учащихся в школе. Дайте этой ячейке имя **Ученики**. Используйте это имя в формулах для вычисления процентов количества учащихся параллели 1-х классов по отношению к общему количеству учащихся школы. Скопируйте эту формулу с использованием маркера заполнения в ячейки для нахождения процентов количества учащихся других параллелей классов по отношению к общему числу учащихся школы.
5. Сохраните книгу в вашей папке в файле с именем **практическая 14.xlsx**.

7.2. НАЗНАЧЕНИЕ И ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКИХ, СТАТИСТИЧЕСКИХ, ЛОГИЧЕСКИХ ФУНКЦИЙ ТАБЛИЧНОГО ПРОЦЕССОРА

1. Какие функции в **Excel** вы знаете? Как вставить функцию в формулу в **Excel**?
2. Какие параметры имеет функция в **Excel**? Что может быть аргументом функции?
3. Как записывается команда ветвления в языке **Object Pascal**? Как она выполняется?

Вы уже знаете, что табличный процессор **Excel 2007** имеет библиотеку встроенных функций, в которую входит более 300 различных функций.

Все они для удобства разделены на категории: *математические*, *статистические*, *текстовые*, *логические*, *финансовые* и др.

В 7 классе вы уже изучали функции в табличном процессоре и рассматривали использование функций **СУММ** – для нахождения суммы чисел, **СРЗНАЧ** – для нахождения среднего арифметического, **МАКС** и **МИН** – для определения максимального и минимального значений в указанном диапазоне ячеек. Первая из них относится к категории математических функций, остальные – к категории статистических.

МАТЕМАТИЧЕСКИЕ ФУНКЦИИ

Рассмотрим еще несколько математических функций (табл. 7.2).

Таблица 7.2

Некоторые математические функции

Функция	*Количество аргументов*	*Результат*	*Пример использования*
ABS(число) (англ. ***absolute*** – абсолютный)	1	Модуль (абсолютная величина) **числа**	**ABS(C10)**
ОКРУГЛ(число; количество разрядов) англ. **ROUND**	2	Округленное **число** Если **количество разрядов** > 0, то число округляется до этого разряда после запятой. Если **количество разрядов** < 0, то число округляется до этого разряда перед запятой. Если **количество разрядов** = 0, то число округляется до целого	**ОКРУГЛ(C1;3)**
КОРЕНЬ(число) англ. **SQRT**	1	Арифметический квадратный корень из **числа**	**КОРЕНЬ(B10)**
СТЕПЕНЬ(число; степень) англ. **POWER**	2	**Число**, возведенное в степень с показателем **степень**	**СТЕПЕНЬ(C5;5)**
ПИ англ. **PI**	0	Приближенное значение числа **π = 3,14159265358979**	**ПИ()**

ЛОГИЧЕСКИЕ ФУНКЦИИ

Кроме формул, состоящих из чисел, знаков действий, функций, в **Excel** могут использоваться логические выражения, содержащие знаки сравнений: > (больше), < (меньше), = (равно), <> (не равно), >= (больше или равно), <= (меньше или равно). Например, **A2+15 = B4-1**; **СУММ(A2:C10)>100** и др.

Такие выражения, как вы знаете, могут принимать значения **ИСТИНА** (англ. *true*) или **ЛОЖЬ** (англ. *false*).

В **Excel** используются и логические функции.

Функцию, результат которой равен ИСТИНА или ЛОЖЬ, называют *логической*.

К **логическим функциям** в **Excel** относятся функции **ЕСЛИ** (англ. *if*), **И** (англ. *and*), **ИЛИ** (англ. *or*), **НЕ** (англ. *not*) и др.

Их используют в формулах тогда, когда должны выполняться различные операции в зависимости от истинности или ложности определенного логического выражения. С такими задачами вы уже встречались, в частности в программировании. Например, когда необходимо начислить работнику премию в размере 20 % от оклада, если он перевыполнил план, а если нет, то не насчитывать. Или если срок продажи товара в магазине закончился, то его уценивают на 50 %.

Общий вид логической функции **ЕСЛИ** следующий:

ЕСЛИ(логическое_выражение;значение_если_истина;значение_если_ложь).

Значение этой функции определяется следующим образом:

1. Вычисляется значение **логического_выражения (Истина** или **Ложь);**
2. Если **логическое_выражение** имеет значение **Истина**, то значение функции равно значению выражения **значение_если_истина;**
3. Если же **логическое_выражение** имеет значение **Ложь**, то значение функции равно значению выражения **значение_если_ложь.**

Например, для функции **ЕСЛИ(A1+B1>100;C1*0,2;C1*0,1)** вычисление значения будет происходить следующим образом. Сначала **Excel** вычислит сумму чисел из ячеек **A1** и **B1** и определит значение логического выражения **A1+B1>100.** Если это значение **Истина**, то результатом вычислений будет значение выражения **C1*0,2.** Если значение логического выражения **A1+B1>100** будет равно **Ложь**, то результат будет вычисляться по формуле **C1*0,1.**

Еще один пример задачи с использованием логической функции **ЕСЛИ:** каждый из рабочих бригады имеет ежедневную норму и реальное количество изготовленных деталей за день. Выяснить для каждого рабочего, выполнил ли он свою норму. На рисунке 7.6 представлено решение этой задачи.

Как видите, правило вычисления логической функции в табличном процессоре аналогично правилам выполнения полной команды ветвления в программировании.

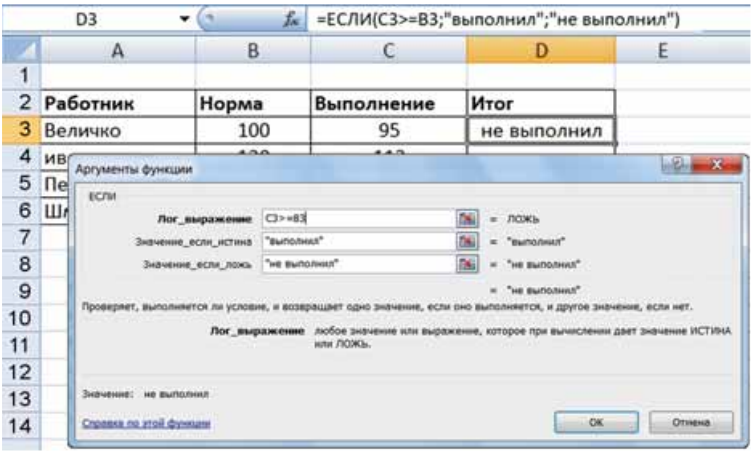


Рис. 7.6. Решение задачи с использованием логической функции **ЕСЛИ**

В **Excel** можно использовать и сокращенную запись логической функции **ЕСЛИ**, которая соответствует команде неполного ветвления:

ЕСЛИ(логическое_выражение;значение_если_истина)

Рассмотрим теперь логические функции **И, ИЛИ, НЕ** (табл. 7.3).

Таблица 7.3

Логические функции И, ИЛИ, НЕ

Функция	*Количество аргументов*	*Результат*	*Пример записи функции*
И(логич1; [логич2];...)	От 1 до 255, все, кроме первого, необязательны	– **Истина**, если все аргументы имеют значение **Истина**; – **Ложь**, если хотя бы один аргумент имеет значение **Ложь**	**И(A1>2; B1>10; B1<20; C1=5)**
ИЛИ(логич1; [логич2];...)	От 1 до 255, все, кроме первого, необязательны	– **Истина**, если хотя бы один аргумент **Истина**; – **Ложь**, если все аргументы имеют значение **Ложь**	**ИЛИ(A1>2; B1>10; B1<20; C1=5)**
НЕ (логич)	1	– **Истина**, если аргумент имеет значение **Ложь**; – **Ложь**, если аргумент имеет значение **Истина**	**НЕ(F1>25)**

Чаще всего эти функции используются как аргументы функции **ЕСЛИ**, а самостоятельно – очень редко.

В таблице 7.4 представлены значения функций **И, ИЛИ, НЕ** в зависимости от данных в ячейках **A1** и **B1**.

Таблица 7.4

Таблица значений функций И, ИЛИ, НЕ

A1	B1	И(A1;B1)	ИЛИ(A1;B1)	НЕ(A1)
Истина	Истина	Истина	Истина	Ложь
Истина	Ложь	Ложь	Истина	Ложь
Ложь	Истина	Ложь	Истина	Истина
Ложь	Ложь	Ложь	Ложь	Истина

Логическую функцию **И** еще называют **конъюнкция**, логическую функцию **ИЛИ – дизъюнкция**, а логическую функцию **НЕ – отрицание**.

Логические функции **И, ИЛИ, НЕ** используют для записи сложных логических выражений. Например, в функции **ЕСЛИ** нельзя записывать двойное неравенство. Поэтому вместо двойного неравенства **10<A3<20** следует использовать логическое выражение **И(A3>10;A3<20)**. Или, например, функцию задано:

$$y = \begin{cases} 2x - 5, \text{ если } x < -2 \text{ или } x > 10, \\ \text{не определена, при других значениях } x \end{cases}$$

и значение x находится в ячейке **A5**, то формула для вычисления значения функции y записывается: **=ЕСЛИ(ИЛИ(A5<–2;A5>10);2*A5–5;"функция не определена»).**

Для тех, кто хочет знать больше

В **Excel 2007** в записи функций можно использовать *вложенные функции* (до 64-х уровней вложенности). Так, второй и третий аргументы функции **ЕСЛИ** могут содержать вложенные арифметические операции и функции, в том числе и функцию **ЕСЛИ**.

Например, **ЕСЛИ(A1<0; –B1/A1; ЕСЛИ(A1>0;B1/A1; «Деление невозможно!»))**.

Самое важное в этом пункте

Табличный процессор **Excel 2007** имеет библиотеку встроенных функций, в которую входит больее 300 различных функций. Все они для удобства разделены на категории: *математические*, *статистические*, *текстовые*, *логические*, *финансовые* и др.

Примеры математических функций: модуль числа **ABS**, арифметический квадратный корень из числа **КОРЕНЬ**, возведение числа в степень **СТЕПЕНЬ**, сумма чисел **СУММ**.

К логическим функциям в **Excel** относятся функции **ЕСЛИ**, **И**, **ИЛИ**, **НЕ** и др. Они используются в формулах тогда, когда следует выполнять различные операции в зависимости от истинности или ложности определенного логического выражения.

Общий вид логической функции **ЕСЛИ**:

ЕСЛИ(логическое_выражение;значение_если_истина;значение_если_ложь).

Значение этой функции определяется следующим образом: вычисляется значение **логического_выражения**, если **логическое_выражение** имеет значение **Истина**, то значение функции равно значению выражения **значение_если_истина**; в противном случае значение функции равно значению выражения **значение_если_ложь**.

Дайте ответы на вопросы

1•. Какой общий вид, количество аргументов, результат выполнения математических функций **ABS**, **ОКРУГЛ**, **ПИ**, **СТЕПЕНЬ**?

2•. Какой общий вид, количество аргументов, результат выполнения математических функций **КОРЕНЬ**, **СУММ**?

3•. Какой общий вид, количество аргументов, результат выполнения статистических функций **СРЗНАЧ**, **МАКС**, **МИН**?

4•. Что такое логические выражения? Какие знаки они могут содержать?

5°. Какие значения может принимать логическое выражение?

6•. Какой общий вид, количество аргументов, результат использования логической функции **ЕСЛИ**?

7•. Как определяется значение логической функции **ЕСЛИ**?

8•. Какой общий вид, количество аргументов, результат выполнения логических функций **И**, **ИЛИ**, **НЕ**?

9•. Как определяется значение логических функций **И, ИЛИ, НЕ**?

10*. Что такое вложенность функций? Какой уровень вложенности максимален?

Выполните задания

1°. Откройте табличный процессор **Excel** 2007. Создайте таблицу для вычисления длин диагоналей 5 прямоугольников по заданным длинам их сторон. Установите формат результатов вычисления с двумя десятичными разрядами. Сохраните книгу в вашей папке в файле с именем **задание 7.2.1.xlsx**.

2•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.2\образец 7.2.2.xlsx**). На листе **Лист1** представлена таблица очков, которые набрал каждый из участников соревнований по пятиборью в каждом виде соревнований. Добавьте в таблицу формулы для вычисления суммы очков и максимального количества очков, набранных каждым из участников, а также среднего количества очков, которые набрали участники в каждом виде соревнований. Среднее количество очков округлите до целых. Сохраните книгу в вашей папке в файле с именем **задание 7.2.2.xlsx**.

3•. В ячейке **A5** содержится число *10*, в ячейке **B5** – число *–7*, а в ячейке **C5** – число *0*. Определите результаты следующих функций:

а) **И(A5>5;A5<20);**

б) **ИЛИ(C5<10;C5>=20);**

в) **ИЛИ(C5<10;A5>20);**

г) **И(A5>–2;B5>0);**

д) **НЕ(B5<20);**

е) **И((B5>5;B5<–5);НЕ(B5<>10));**

ж) **ИЛИ(И(A5>2;A5<=10);B5<0);**

з) **НЕ(И(A5<100;C5=0)).**

4°. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.2\образец 7.2.4.xlsx**). На листе **Лист1** представлено количество деталей, которые сделал каждый из рабочих в течение дня. В ячейке **A1** содержится дневная норма плана. Добавьте в таблицу формулы для определения, выполнил каждый из рабочих норму или нет. Результат указать в виде **Истина** или **Ложь**. Сохраните книгу в вашей папке в файле с именем **задание 7.2.4.xlsx**.

5*. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.2\образец 7.2.5.xlsx**). На листе **Лист1** представлены результаты выступлений участников олимпиады в каждом из двух туров. По условиям проведения олимпиады *Дипломом I степени* награждаются участники, набравшие не менее 85 % очков, *Дипломом II степени* – участники, набравшие не менее 75 %, но меньше 85 % очков, *Дипломом III степени* – участники, набравшие не менее 50 %, но меньше 75 % очков, остальные получают Диплом участника. Добавьте в таблицу формулы для определения, какой диплом будет вручен каждому из участников. Сохраните книгу в вашей папке в файле с именем **задание 7.2.5.xlsx**.

6*. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.2\образец 7.2.6.xlsx**). На листе **Лист1** представлена таблица зна-

чений переменной x. Добавьте в таблицу формулы для определения соответствующих значений функции

$$y = \begin{cases} 2x - 5, \text{ если } x < -2 \text{ или } x > 10, \\ 3x + 1, \text{ если } 2 \leqslant x \leqslant 3, \\ x^5 - 4x^3 \text{ при других значениях } x. \end{cases}$$

Сохраните книгу в вашей папке в файле с именем **задание 7.2.6.xlsx**.

7*. Откройте табличный процессор **Excel 2007**. Составьте таблицу, по которой определите, какие условия вклада в банк выгоднее клиенту: 24 % годовых с начислением процентов ежемесячно или 26 % годовых с начислением процентов каждые полгода. Сохраните книгу в вашей папке в файле с именем **задание 7.2.7.xlsx**.

7.3. ПОСТРОЕНИЕ И ОБРАБОТКА ДИАГРАММ РАЗЛИЧНОГО ТИПА

1. Что такое диаграммы? Для чего они предназначены?
2. Какие объекты диаграмм вы знаете? Какие свойства они имеют?
3. Как создать диаграмму? Какие операции редактирования и форматирования можно выполнять над диаграммой?

ДИАГРАММЫ В EXCEL 2007

Из курса информатики седьмого класса вы уже знаете, что для более наглядного представления и анализа числовых данных используют диаграммы, на которых соотношение между числами отображается с помощью геометрических фигур. Размеры этих фигур на диаграммах пропорциональны числовым данным, по которым эти диаграммы построены.

Чаще всего используют круговые диаграммы и гистограммы, с которыми вы уже ознакомились в седьмом классе. **Круговые** диаграммы предназначены для отображения доли, которую составляет каждое число в их общей сумме. **Гистограмму** целесообразно создавать тогда, когда нужно сравнить значения одного или нескольких наборов числовых данных.

Кроме этих типов диаграмм, в **Excel 2007** можно построить диаграммы и других типов, например: *линейчатую*, *график*, *точечную*, *пузырьковую*, *лепестковую* и др. Каждый из этих типов имеет несколько разновидностей. Их перечень можно просмотреть, открыв списки соответствующих кнопок на вкладке **Вставка** в группе **Диаграммы**.

Рассмотрим подробнее некоторые из них.

Линейчатая диаграмма аналогична гистограмме, только геометрические фигуры расположены горизонтально (рис. 7.7). Такие диаграммы удобно использовать, если горизонтальные элементы данных нагляднее, чем вертикальные, например, если вдоль горизонтальной оси откладываются возрастающие промежутки времени, температуры и т. п. Их удобно использовать еще и в тех случаях, когда подписи вдоль вертикальной оси длинные и легче воспринимаются в горизонтальном положении.

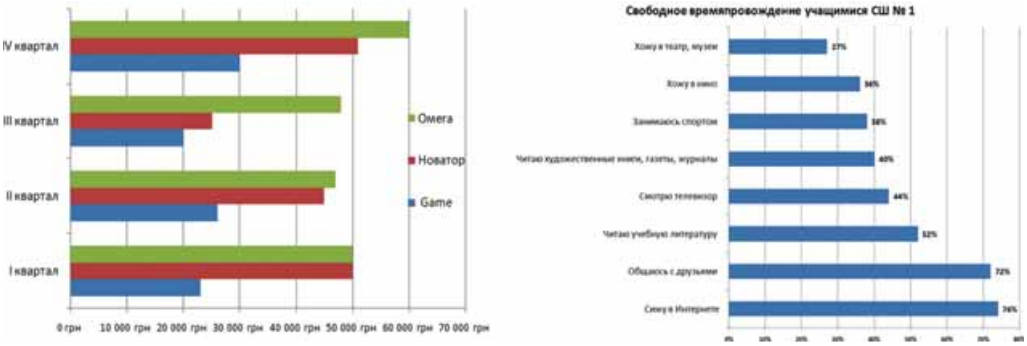


Рис. 7.7. Линейчатые диаграммы

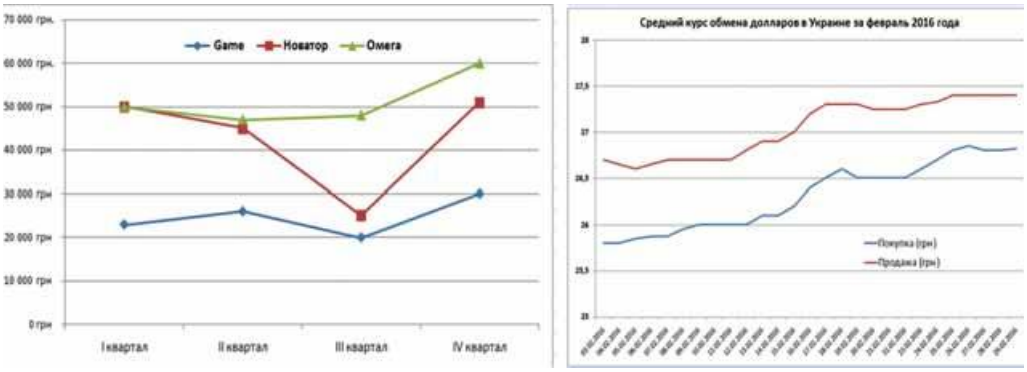


Рис. 7.8. График с маркерами данных и без них

	A	B	C	D	E	F	G	H	I	J
1	Зависимость спроса Y на товар от его цены X выражается формулой Y=2/X+2									
2										
3	Цена товара (грн)	Спрос на товар								
4	100	2,02								
5	150	2,01								
6	200	2,01								
7	250	2,01								
8	300	2,01								
9	350	2,01								
10	400	2,01								
11	450	2,00								
12	500	2,00								
13	550	2,00								
14	600	2,00								
15										
16										

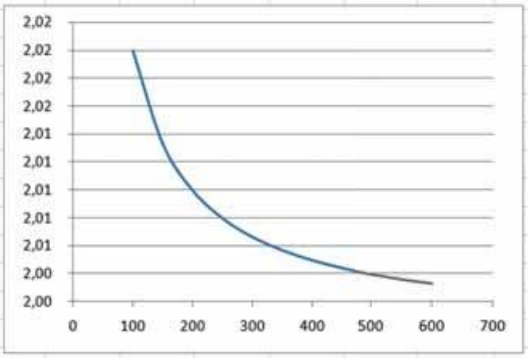


Рис. 7.9. Точечная диаграмма

Графики целесообразно использовать, если нужно отобразить некоторый процесс во времени, динамику изменения данных, которые не зависят друг от друга (рис. 7.8). График можно строить с маркерами и без них (маркеры отображают на диаграмме точку данных).

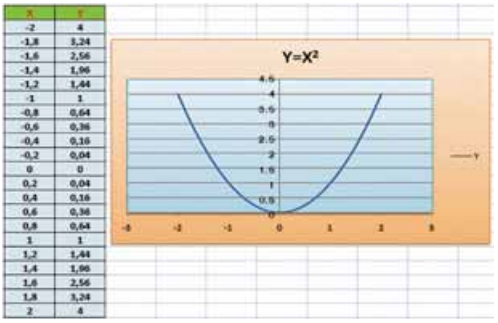


Рис. 7.10. График функции



Рис. 7.11. Графики двух функций

Точечные диаграммы используют для демонстрации взаимосвязи между двумя переменными, когда значение одной величины зависит от значения другой (рис. 7.9).

Точечные диаграммы чаще всего используют для построения графиков функций (рис. 7.10). На одной диаграмме можно построить графики двух функций и использовать их для приближенного решения уравнений (рис. 7.11).

ПОСТРОЕНИЕ ДИАГРАММ

Как вы знаете из седьмого класса, диаграммы строятся на основе числовых данных, представленных в электронной таблице. Диаграммы являются динамическими – с изменением данных в таблице диаграммы автоматически изменяются.

Создать диаграмму можно, используя элементы управления вкладки **Вставка** группы **Диаграммы**. Для этого необходимо:

1. Выделить диапазон ячеек с данными, на основе которых будет строиться диаграмма. *Примечание:* выделенный диапазон ячеек может быть как связным, так и несвязным. Желательно, чтобы в него вошли подписи соответствующих строк и столбцов.
2. Открыть на вкладке **Вставка** в группе **Диаграммы** список кнопки нужного типа диаграмм.
3. Выбрать вид диаграммы.

В результате выполнения указанного алгоритма на листе рядом с электронной таблицей будет построена диаграмма, значения свойств объектов которой будет устанавлено по умолчанию.

На диаграмме выделяют следующие объекты: область диаграммы, область построения диаграммы, заголовок диаграммы, элемент данных, ряд данных, легенда, главная горизонтальная ось, главная вертикальная ось, названия осей, подписи данных.

РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ ДИАГРАММ

Из курса информатики седьмого класса вы знаете, что редактирование диаграммы заключается в следующем:

- изменении диапазона данных, по которым строится диаграмма: обмен местами столбцов и строк на диаграмме, добавление новых строк

(столбцов) данных в уже существующие диапазоны построения диаграмм или удаление некоторых из них;
- вставке или изменении названий диаграммы и ее осей, легенды, подписей данных и др.;
- наличии осей и линий сетки;
- выделении и редактировании отдельных объектов диаграммы;
- изменении типа диаграммы;
- изменении местоположения диаграммы (перемещение на листе, размещение на отдельном листе)

и другое.

Форматирование диаграммы предусматривает:
- изменение размеров диаграммы;
- изменение стиля оформления диаграммы;
- изменение шрифта, размера, цвета, заливки и т. д. отдельных объектов диаграммы;
- настройку отображения осей и сетки на диаграмме

и другое.

Для выполнения операций редактирования и форматирования используют элементы управления вкладок **Конструктор**, **Макет** и **Формат** временного раздела **Работа с диаграммами**.

Напомним порядок выполнения некоторых операций редактирования диаграмм:
- чтобы изменить тип диаграммы, следует выбрать ее и выполнить ***Конструктор*** ⇒ ***Тип*** ⇒ ***Изменить тип диаграммы***;
- чтобы поменять местами отображение столбцов и строк, следует выбрать диаграмму и выполнить ***Конструктор*** ⇒ ***Данные*** ⇒ ***Строка/Столбец***;
- чтобы изменить размеры диаграммы, следует перетащить один из маркеров изменения размеров, которые расположены на ее границе;
- чтобы изменить местоположение диаграммы на листе, ее следует перетащить в нужное место.

Как вы уже знаете из седьмого класса, для форматирования объекта диаграммы его сначала следует выбрать, а потом можно:
- выбрать один из стилей оформления выделенного объекта из списка группы **Стили фигур** вкладки **Формат**;
- выбрать один из стилей оформления текстового объекта из списка группы **Стили WordArt** вкладки **Формат**;
- выбрать один из видов заливки, оформления границ, эффектов из списков кнопок **Заливка фигуры**, **Контур фигуры**, **Эффекты для фигур** группы **Стили фигур** вкладки **Формат**;
- выполнить **Формат** ⇒ ***Текущий фрагмент*** ⇒ ***Формат выделенного фрагмента*** и на вкладках окна установить необходимые значения свойств выделенного объекта;
- использовать команды контекстного меню выделенного объекта.

Самое важное в этом пункте

Линейчатые диаграммы аналогичны гистограммам, и используются, если горизонтальные элементы данных нагляднее, чем вертикальные.

Графики целесообразно строить, если необходимо отобразить непрерывные данные, а также динамику изменения независимых между собой данных.

Точечные диаграммы используют для графического отображения зависимых между собой данных, чаще всего для построения графиков функций.

Создать диаграмму можно, используя элементы управления вкладки **Вставка** группы **Диаграммы**.

Для выполнения операций редактирования и форматирования выделенного объекта диаграмм используют элементы управления вкладок **Конструктор**, **Макет** и **Формат** временного раздела **Работа с диаграммами**.

Дайте ответы на вопросы

1°. Какие вы знаете виды диаграмм в **Excel 2007**?
2°. Какие объекты есть на диаграмме в **Excel 2007**? Опишите их.
3°. Какие свойства имеют объекты диаграмм в **Excel 2007**?
4•. Как создать *линейчатую диаграмму*? Каково ее назначение?
5•. Как создать *график*? Каково его назначение?
6•. Как создать *точечную диаграмму*? Каково ее назначение?
7•. Чем отличается использование диаграммы *график* от использования *точечной диаграммы*?
8*. Как можно использовать точечные диаграммы для приближенного решения уравнений?
9•. Какие операции редактирования диаграмм вы знаете? Как их выполнять?
10•. Какие операции форматирования диаграмм вы знаете? Как их выполнять?
11•. Как определить тип диаграммы, который необходимо применить?

Выполните задание

1•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.3\образец 7.3.1.xlsx**). На листе **Лист1** представлена таблица осадков за последний месяц. Постройте по данным этой таблицы круговую диаграмму, линейчатую диаграмму и график. Какая из диаграмм нагляднее демонстрирует данные? Оформите эту диаграмму (название, подписи данных, формат текста, заливка и т. п.) и разместите ее на отдельном листе. Сохраните книгу в вашей папке в файле с именем **задание 7.3.1.xlsx**.

2•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.3\образец 7.3.2.xlsx**). На листе **Лист1** представлена таблица добычи газа в Украине разными компаниями за три последних года. По данным этой таблицы постройте линейчатую диаграмму добычи газа за последний год разными компаниями и график суммарной добычи газа за последние три года. Разместите диаграммы на отдельных листах. Оформите диаграммы (название, подписи осей и данных, формат текста, заливка и т. д.). Сохраните книгу в вашей папке в файле с именем **задание 7.3.2.xlsx**.

3•. Найдите в Интернете данные о количестве населения в вашем населенном пункте (или в Украине) за последние 5 лет. Создайте элек-

тронную таблицу по этим данным. Постройте по данным таблицы три диаграммы: круговую, линейчатую и график. Разместите их на отдельных листах и оформите по своему усмотрению. Сохраните книгу в вашей папке в файле с именем **задание 7.3.3.xlsx**.

4•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.3\образец 7.3.4.xlsx**). На листе **Лист1** представлена таблица растворимости некоторых веществ в зависимости от температуры. Постройте по данным этой таблицы точечную диаграмму с гладкими линиями и маркерами. Добавьте заголовок, подписи осей и данных, установите один из стилей оформления. Сохраните книгу в вашей папке в файле с именем **задание 7.3.4.xlsx**.

5•. Откройте табличный процессор **Excel 2007**. На листе **Лист1** создайте таблицу и постройте график функции $y = x^3 - 3x$ на отрезке [–3; 3] с шагом 0,2. Сохраните книгу в вашей папке в файле с именем **задание 7.3.5.xlsx**.

6•. Откройте табличный процессор **Excel 2007**. На листе **Лист1** создайте таблицу и постройте график функции $y = \dfrac{1}{x^2 + 1}$ на отрезке [–3; 3] с шагом 0,2. Сохраните книгу в вашей папке в файле с именем **задание 7.3.6.xlsx**.

7*. Откройте табличный процессор **Excel 2007**. На листе **Лист1** создайте таблицу и постройте графики функций $y = x^4 - 2x^2$ и $y = 1/4x - 0{,}5$ на отрезке [–3; 3] с шагом 0,2 и определите приближенные значения корней уравнения $x^4 - 2x^2 = 1/4x - 0{,}5$ на этом отрезке. Сохраните книгу в вашей папке в файле с именем **задание 7.3.7.xlsx**.

8*. Откройте табличный процессор **Excel 2007**. Выясните, используя **Справку**, особенности назначения и построения пузырьковой и лепестковой диаграмм. Найдите в Интернете дневник погоды по вашему областному центру. Создайте по найденным данным электронную таблицу облачности за предыдущий месяц: количество ясных, малооблачных, облачных и пасмурных дней. Постройте по данным этой таблицы пузырьковую и лепестковую диаграммы. Разместите диаграммы рядом под таблицей. Оформите их по своему усмотрению. Сохраните книгу в вашей папке в файле с именем **задание 7.3.8.xlsx**.

ПРАКТИЧЕСКАЯ РАБОТА № 15

«Использование математических, логических и статистических функций табличного процессора. Построение диаграмм»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте указанный учителем файл (например, **Глава7\Практическая15\образец пр15.xlsx**).
2. На листе **Лист1** представлена таблица показателей счетчика электроэнергии за каждый месяц прошлого года. В ячейке **A1** находится стоимость **1 кВт·ч**. Добавьте в таблицу формулы для вычисления

ежемесячной платы за электроэнергию, суммарной платы за год, средних ежемесячных расходов электроэнергии.

3. Постройте на отдельном листе график ежемесячной платы за электроэнергию. Добавьте название диаграммы, подписи осей и значений. Отформатируйте диаграмму по своему усмотрению.
4. На листе **Лист2** представлены фамилии учащихся класса и их рост. Для каждого ученика класса определите, больше ли его рост среднего роста по классу.
5. Постройте на отдельном листе линейчатую диаграмму роста учащихся класса. Добавьте название диаграммы, подписи осей и значений. Отформатируйте диаграмму по своему усмотрению.
6. На листе **Лист3** составте таблицу значений функции $y = x^4 - 4x$ на отрезке [–3; 3].
7. Постройте на отдельном листе график этой функции, используя точечную диаграмму. Добавьте название диаграммы, подписи осей и значений. Отформатируйте диаграмму по своему усмотрению.
8. Сохраните книгу в своей папке в файле с именем **практическая 15.xlsx**.

7.4. СОРТИРОВКА ДАННЫХ В ТАБЛИЦАХ. АВТОМАТИЧЕСКИЕ И РАСШИРЕННЫЕ ФИЛЬТРЫ

1. Какой порядок расположения символов в таблице **Unicode**?
2. Какие значения логических функций **И**, **ИЛИ** в зависимости от значений аргументов?
3. Каково назначение символов * и ? в масках имен файлов?

СОРТИРОВКА ДАННЫХ В ЭЛЕКТРОННОЙ ТАБЛИЦЕ

Данные в ячейках электронной таблицы можно **упорядочивать (сортировать), то есть изменять порядок их расположения в строках или столбцах по некоторым признакам.**

В табличном процессоре можно выполнять два вида сортировки: **по возрастанию** или **по убыванию.** Порядок сортировки зависит от типа данных, осуществляется посимвольно, в порядке ***размещения символов в кодовой таблице Unicode.***

В **Excel 2007** сортировка по возрастанию происходит по следующим правилам:

- если в таблице находятся данные различных типов, то после сортировки они будут размещаться в следующем порядке: числовые, дата/время, текстовые, логические;
- числа упорядочиваются от наименьшего значения к наибольшему;
- даты упорядочиваются от самой старой к самой новой;
- сначала располагаются тексты, написанные латиницей, затем – кириллицей;

Введенные данные	Отсортированные по возрастанию	Отсортированные по убыванию
35	-10	ИСТИНА
школа	0,5	ЛОЖЬ
-10	35	школа
	Lyceum	Сиренко
ИСТИНА	Сиренко	Lyceum
	школа	35
0,5	ЛОЖЬ	0,5
ЛОЖЬ	ИСТИНА	-10
Lyceum		
Сиренко		

Рис. 7.12. Пример сортировки данных

- сначала размещаются тексты, написанные прописными буквами, потом – заглавными;
- тексты упорядочиваются так: сначала сравниваются первые символы, в случае их равенства – вторые символы и т. д.;
- логическое значение **ЛОЖЬ** размещается перед значением **ИСТИНА**;
- пустые ячейки располагаются последними (хотя символ **Пробел** в таблице **Unicode** стоит первым и имеет наименьший код).

При сортировке по убыванию порядок расположения будет обратным.

Примеры размещения значений различных типов после сортировки по возрастанию и по убыванию представлены на рисунке 7.12.

Для того чтобы быстро выполнить сортировку строк некоторого диапазона по значениям первого столбца этого диапазона, следует:

1. Выделить нужный диапазон ячеек.
2. Выполнить ***Главная*** ⇒ ***Редактирование*** ⇒ ***Сортировка и фильтр*** ⇒ ***Сортировка от минимального к максимальному*** или ***Сортировка от максимального к минимальному*** (рис. 7.13).

В результате этих действий строки выделенного диапазона меняют свое расположение так, что значения в первом столбце этого диапазона будут упорядочены в выбранном порядке, а данные в других столбцах будут перемещены вместе с соответствующей строкой.

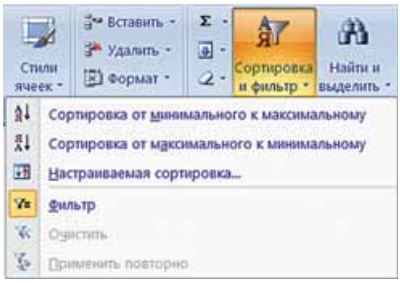


Рис. 7.13. Список команд кнопки **Сортировка и фильтр**

Обращаем ваше внимание:

1. Названия команд сортировки автоматически меняются в зависимости от типа данных, подлежащих упорядочиванию:
   - для числовых смешанных данных – **Сортировка от минимального к максимальному** и **Сортировка от максимального к минимальному;**
   - для текстовых данных – **Сортировка от А до Я** и **Сортировка от Я до А;**
   - для даты и времени – **Сортировка от старших к новым** и **Сортировка от новых к старшим.**
2. Если выполнять сортировку, виделив даные только в одном столбце, открывается окно **Обнаружены данные вне указанного диапазона** (рис. 7.14), в котором предлагается автоматически расширить выделенный диапазон ячееек или сортировать в пределах указанного выделения.

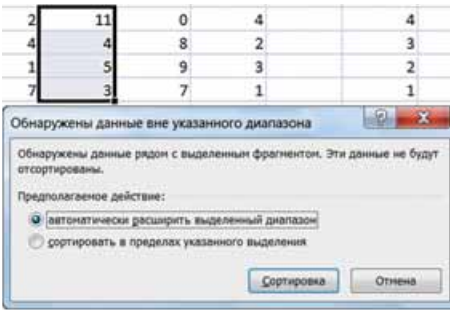


Рис. 7.14. Окно **Обнаружены данные вне указанного диапазона**

Сортировку в выделенном диапазоне можно осуществлять и по значениям нескольких столбцов. Это происходит так:

- сначала строки сортируются по значениям в первом из указанных столбцов;

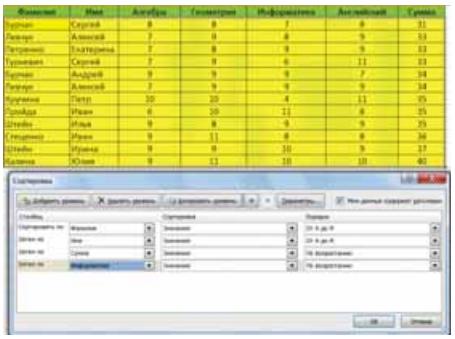


Рис. 7.15. Кнопка **Сортировка**, диапазон ячеек для сортировки и окно **Сортировка**

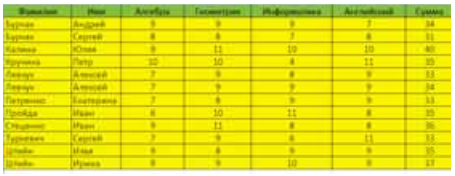


Рис. 7.16. Диапазон ячеек, отсортированный по условиям, поданным на рисунке 7.15

- затем сортировка в каждом следующем из выбранных столбцов происходит лишь для тех строк, в которых значения во всех предыдущих указанных для сортировки столбцах совпадают.

Каждый из указанных для сортировки столбцов называют *уровнем сортировки*.

Объясним такую сортировку на конкретном примере. На рисунке 7.15 приведен пример таблицы годовых оценок учащихся школы, а на рисунке 7.16 эту же таблицу отсортировано по значениям в четырех столбцах: **Фамилия** (*от А до Я*), **Имя** (*от А до Я*), **Сумма** (*от наименьшего к наибольшему*) и **Информатика** (*от наименьшего к наибольшему*).

Процесс сортировки происходил так. Сначала строки переставлялись так, чтобы даные в столбце **Фамилия** были расположены в алфавитном порядке (**от А до Я**). И если бы у всех учащихся фамилии были разные, то больше никакой сортировки этой таблицы не было бы. Но таблица содержит три группы учащихся с одинаковыми фамилиями: *Бурчак*, *Левчук*, *Штейн*.

Дальнейшая сортировка происходит уже по значениям во втором из выбранных столбцов (**Имя**) и только в пределах каждой из этих трех групп строк, по каждой отдельно. В результате этого шага сортировки переставились (а может и нет) первая и вторая строки, пятая и шестая строки, одиннадцатая и двенадцатая строки так, чтобы имена были расположены в алфавитном порядке (**от А до Я**).

Сортировка по значениям в третьем из выбранных столбцов (**Сумма**) происходила только в пределах пятой и шестой строк, в которых значения в первых из двух выбранных столбцов совпали (*Левчук Алексей*). Сортировки по значениям в четвертом из выбранных столбцов не происходило, потому что не оказалось хотя бы двух строк, в которых значения в первых из трех выбранных для сортировки столбцов (**Фамилия**, **Имя**, **Сумма**) совпали.

Чтобы отсортировать строки в выделенном диапазоне ячеек по значениям в нескольких столбцах, следует:

1. Выделить нужный связанный диапазон ячеек (желательно, чтобы в него вошли подписи строк, но к нему не должны входить объединенные ячейки).
2. Выполнить ***Данные*** ⇒ ***Сортировка и фильтр*** ⇒ ***Сортировка***.
3. Установить или изменить значения параметров сортировки: в списке поля **Сортировать по** выбрать имя первого столбца, по значениям в

котором нужно отсортировать строки, в списке поля **Сортировка по** – *Значение*, а в списке поля **Порядок** – порядок сортировки (рис. 7.15).

4. Определить условия сортировки по значениям в другом столбце, выбрав кнопку **Добавить уровень** или **Копировать уровень**, и установить в полях строки **Затем** значения параметров сортировки по значениям во втором выбранном столбце.
5. При необходимости повторить команду 4 несколько раз.
6. Если в выделенный диапазон вошла строка заголовка, но отметка флажка **Мои данные содержат заголовки** не установлена, установить ее (тогда строка заголовка НЕ БУДЕТ учитываться при сортировке и останется на своем месте).
7. Выбрать кнопку **ОК**.

Для тех, кто хочет знать больше

Сортировать можно не только строки, но и столбцы. По умолчанию в **Excel** устанавливается режим сортировки строк по значениям в столбцах. Для изменения этого режима сортировки следует в окне **Сортировка** выбрать кнопку **Параметры**, а затем – соответствующий переключатель в группе **Сортировать**. Там же можно определить, учитывать ли регистр при сортировке, то есть, различать ли прописные и заглавные буквы.

АВТОМАТИЧЕСКИЕ ФИЛЬТРЫ

***Фильтрация* – это выбор значений в ячейках электронной таблицы, соответствующих определенным условиям.**

После выполнения фильтрации в таблице отображаются только те строки, данные в которых соответствуют **условиям фильтрации**. Остальные – временно скрываются.

Чтобы установить режим фильтрации данных, следует:

1. Установить табличный курсор в произвольную ячейку таблицы или выделить необходимый диапазон ячеек.
2. Выполнить ***Главная*** ⇒ ***Редактирование*** ⇒ ***Сортировка и фильтр*** ⇒ ⇒ ***Фильтр*** (или ***Данные*** ⇒ ***Сортировка и фильтр*** ⇒ ***Фильтр***).

После выполнения этих действий у правой границы каждой ячейки первой строки таблицы или выделенного диапазона ячеек появятся кнопки открытия списка команд фильтрации (рис. 7.17). В этих списках находятся команды сортировки, фильтрации по цвету, а также числовые и текстовые фильтры.

Рассмотрим некоторые примеры выполнения фильтрации.

Чаще всего выполняют **фильтрацию по значениям** в некотором столбце данных. Например, по представленной таблице оценок отобразить список учеников, которые имеют по информатике оценку 9. Для этого следует:

1. Открыть список команд фильтрации в столбце **Информатика**.
2. Снять отметку флажка **Выделить все**.

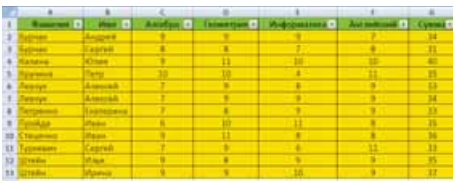


Рис. 7.17. Диапазон ячеек после выполнения команды **Фильтр**

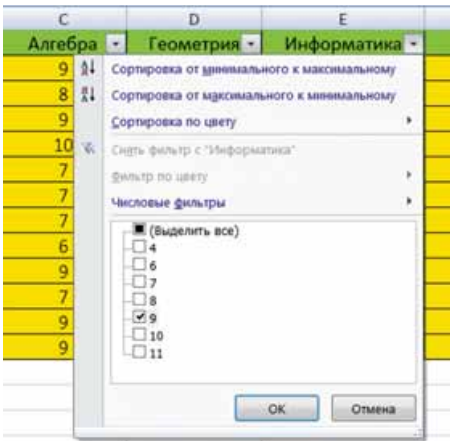


Рис. 7.18. Применение фильтра по столбцу **Информатика**

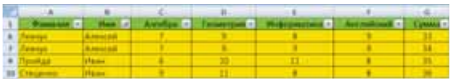


Рис. 7.21. Результат применения фильтра на рисунке 7.20

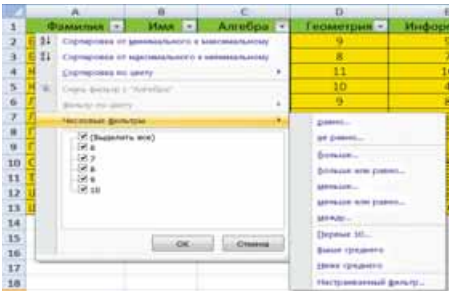


Рис. 7.22. Команды установки условий фильтрации числовых данных

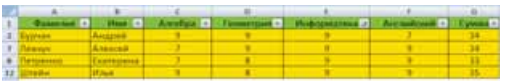


Рис. 7.19. Результат применения фильтра на рисунке 7.18

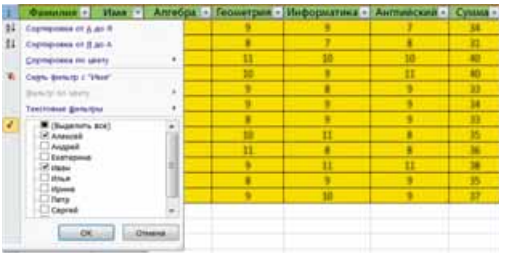


Рис. 7.20. Фильтрация по двум значениям в столбце **Имя**

3. Установить отметку флажка у значения **9**.
4. Выбрать кнопку **ОК**.

После выполнения этих действий в таблице будут отображены только те строки, значение в которых в нужном столбце равны указанным. Выполнение и результат описанного примера представлены на рисунках 7.18 и 7.19.

Обратите внимание: вид кнопки фильтра в столбце **Информатика** на рисунке 7.19 изменился на такой  . Наличие такой кнопки в ячейке означает наличие установленного фильтра для данных в этом столбце таблицы.

Фильтрацию можно осуществлять в столбце по нескольким значениям. Например, нужно отобразить данные об учениках с именами, *Иван* или *Алексей*. Для этого следует:

1. Открыть список команд фильтрации в столбце **Имя**.
2. Снять отметку флажка **Выделить все**.
3. Установить отметку флажка у значений *Иван* и *Алексей* (рис. 7.20).
4. Выбрать кнопку **ОК**.

В результате получим таблицу, которая представлена на рисунке 7.21.

Если в столбце находятся только числовые данные, то по данным этого столбца можно выполнять фильтрацию с использованием команды **Числовые фильтры**, применяя команды **равно**, **не равно**, **больше**, **меньше**, **между** и т. п. (рис. 7.22).

Например, нужно отобразить данные о тех учениках, которые имеют суммарный балл по предметам $\geqslant$ **35** и $\leqslant$ **40**. Для этого следует выполнить:

1. Открыть список команд фильтрации в столбце **Сумма**.
2. Выбрать ***Числовые фильтры*** ⇒ ***между***.
3. Ввести в соответствующие поля окна необходимые данные – **35** и **40** (рис. 7.23).
4. Выбрать кнопку **ОК**.

Применив фильтр на рисунке 7.23, получим таблицу, как на рисунке 7.24.

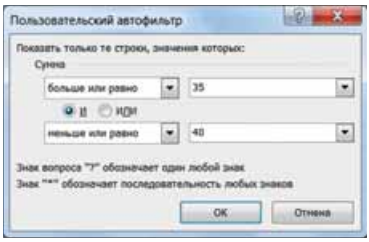
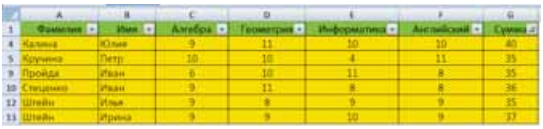


Рис. 7.23. Установка числового фильтра

Еще один числовой фильтр, команда **Первые 10** открывает окно **Наложение условия по списку** (рис. 7.25), в котором можно установить условие фильтрации, по которому в таблице будут отображаться только те строки, которые содержат в данном столбце 10 (или другое установленное количество) наибольших (наименьших) значений.

Команда **выше среднего (нижее среднего)** выполняет фильтрацию, по которой в таблице отображаются только те строки, значения в которых в данном столбце, большие (меньшие) чем среднее арифметическое значение в данном столбце.



	A	B	C	D	E	F	G
1	Фамилия	Имя	Алгебра	Геометрия	Информатика	Английский	Сумма
4	Калина	Юлия	9	11	10	10	40
5	Кручина	Петр	10	10	4	11	35
9	Пройда	Иван	6	10	11	8	35
10	Стеценко	Иван	9	11	8	8	36
12	Шейн	Илья	9	8	9	9	35
13	Шейн	Ирина	9	9	10	9	37

Рис. 7.24. Результат применения фильтра на рисунке 7.23



Рис. 7.25. Окно установки значения для команды **Первые 10...**

Для текстовых данных применяются несколько иные команды фильтрации (рис. 7.26): **равно, не равно, начинается с, заканчивается на, содержит** и др.

Например, нужно отобразить данные о тех учениках, фамилии которых начинаются с буквы ***П***. Для этого следует выполнить:

1. Открыть список команд фильтрации в столбце **Фамилия**.
2. Выбрать ***Текстовые фильтры*** ⇒ ***начинается с...***
3. Ввести в соответствующее поле букву ***П*** (рис. 7.27).
4. Выбрать кнопку **ОК**.



Рис. 7.26. Команды фильтрации текстовых данных

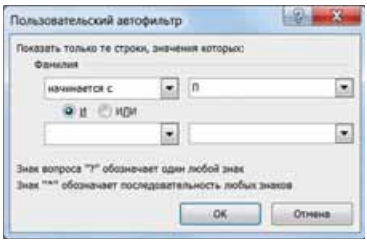


Рис. 7.27. Условие фильтрации текстов

Фильтрацию можно выполнять последовательно для нескольких столбцов таблицы. Тогда каждый следующий фильтр будет применяться уже только для тех строк таблицы, которые отображаются после применения всех предыдущих фильтров. То есть *общее условие фильтрации – это условия фильтрации для каждого столбца, объединенные логической операцией* ***И***. Например, если провести фильтрацию по данным столбца **Сумма** с условием **> 35**, а затем по данным столбца **Информатика** с условием **> 8**, то общее условие фильтрации будет таким – **(*сумма баллов* > 35) И (*балл по информатике* > 8)**. Результат такой фильтрации приведен на рисунке 7.28.

Отменить фильтрацию данных можно несколькими способами:

- выполнить ***Главная*** ⇒ ***Редактирование*** ⇒ ***Сортировка и фильтр*** ⇒ ***Очистить***;
- выполнить команду **Снять фильтр с** в списке столбца, по данным которого была выполнена фильтрация;
- установить отметку флажка **Выделить все** в списке столбца, по данным которого была выполнена фильтрация, после чего выбрать кнопку **ОК**.

Чтобы вообще отменить режим, в котором можно проводить фильтрацию, следует повторно выполнить одну из команд, которая устанавливает этот режим.

	A	B	C	D	E	F	G
1	Фамилия	Имя	Алгебра	Геометрия	Информатика	Английский	Сумма
4	Калина	Юлия	9	11	10	10	40
13	Штейн	Ирина	9	9	10	9	37

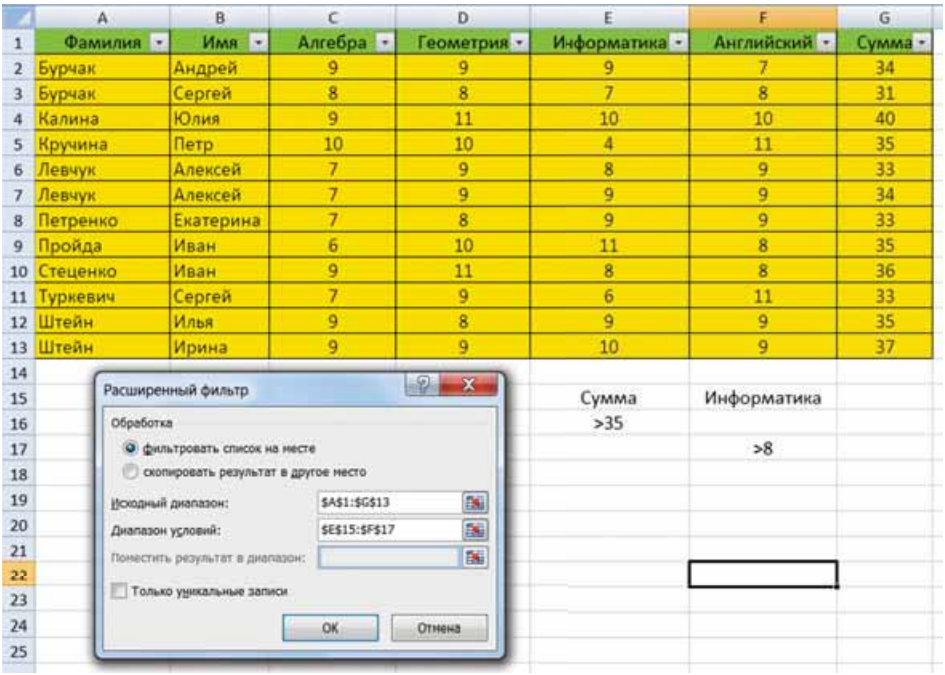
Рис. 7.28. Результат последовательной фильтрации по значениям столбцов **Сумма** и **Информатика**

ДОПОЛНИТЕЛЬНЫЕ ФИЛЬТРЫ

Использование автофильтров дает возможность выполнить фильтрацию не во всех случаях. Так, например, рассмотренными способами нельзя выполнить фильтрацию при условии, которое является объединением условий фильтрации двух столбцов логической операцией **ИЛИ**, например **(сумма баллов больше 35) ИЛИ (балл по информатике больше 8)**.

Выполнить такие виды фильтрации в табличном процессоре **Excel 2007** можно с помощью **расширенных фильтров**. Для этого следует:

1. Скопировать или ввести в свободные ячейки электронной таблицы названия тех столбцов, по которых будет осуществляться фильтрация (например, на рисунке 7.29 названия столбцов **Сумма**, **Информатика**).
2. Ввести в ячейки под названиями столбцов условия фильтрации. Если эти условия должны объединяться логической операцией **И**, то условия фильтрации должны располагаться в одной строке. Если логической операцией **ИЛИ** – то в разных (на рисунке 7.29 условия **>35**, **>8**).
3. Выполнить ***Данные*** ⇒ ***Сортировка и фильтр*** ⇒ ***Дополнительно***.
4. В окне **Расширенный фильтр**:



	A	B	C	D	E	F	G
1	Фамилия	Имя	Алгебра	Геометрия	Информатика	Английский	Сумма
2	Бурчак	Андрей	9	9	9	7	34
3	Бурчак	Сергей	8	8	7	8	31
4	Калина	Юлия	9	11	10	10	40
5	Кручина	Петр	10	10	4	11	35
6	Левчук	Алексей	7	9	8	9	33
7	Левчук	Алексей	7	9	9	9	34
8	Петренко	Екатерина	7	8	9	9	33
9	Пройда	Иван	6	10	11	8	35
10	Стеценко	Иван	9	11	8	8	36
11	Туркевич	Сергей	7	9	6	11	33
12	Штейн	Илья	9	8	9	9	35
13	Штейн	Ирина	9	9	10	9	37

Рис. 7.29. Условия фильтрации и окно **Расширенный фильтр**

а) Выбрать один из переключателей для определения области расположения результата фильтрации (на рисунке 7.29 – фильтровать список на месте).

б) Ввести в поле **Исходный диапазон** диапазон ячеек, данные в котором должны фильтроваться (на рисунке 7.29 – **B2:H14**).

в) Ввести в поле **Диапазон условий** диапазон ячеек, в которых расположены названия столбцов и условия фильтрации.

г) Если был выбран переключатель **скопировать результат в другое место**, то ввести в соответствующее поле диапазон ячеек, где должен разместиться результат фильтрации.

5. Выбрать кнопку **ОК**.

На рисунке 7.30 представлен результат фильтрации, выполненной по условиям, представленным на рисунке 7.29.

	A	B	C	D	E	F	G
1	Фамилия	Имя	Алгебра	Геометрия	Информатика	Английский	Сумма
2	Бурчак	Андрей	9	9	9	7	34
4	Калина	Юлия	9	11	10	10	40
7	Левчук	Алексей	7	9	9	9	34
8	Петренко	Екатерина	7	8	9	9	33
9	Пройда	Иван	6	10	11	8	35
10	Стеценко	Иван	9	11	8	8	36
12	Штейн	Илья	9	8	9	9	35
13	Штейн	Ирина	9	9	10	9	37

Рис. 7.30. Результат фильтрации по условиям, представленным на рисунке 7.29

Самое важное в этом пункте

Данные в ячейках электронной таблицы можно **сортировать, то есть изменять порядок их расположения в строках или столбцах по некоторым признакам.** Сортировка может проводиться **по возрастанию** или **по убыванию.**

Для того чтобы быстро выполнить сортировку выделенного связанного диапазона по значениям первого столбца этого диапазона, следует выполнить ***Главная*** ⇒ ***Редактирование*** ⇒ ***Сортировка и фильтр*** ⇒ ***Сортировка от минимального к максимальному*** или ***Сортировка от максимального к минимальному.***

Сортировку в выделенном диапазоне можно осуществлять и по значениям нескольких столбцов. Для этого следует выполнить ***Данные*** ⇒ ***Сортировка и фильтр*** ⇒ ***Сортировка*** и установить параметры и условия сортировки в соответствующем окне. При этом сначала строки сортируются по значениям в первом из указанных столбцов, а затем в каждом следующем из выбранных столбцов сортировка происходит только для тех строк, в которых значения во всех предыдущих указанных для сортировки столбцах совпадают.

Фильтрация **– это выбор значений в ячейках электронной таблицы, соответствующих определенным условиям.** После выполнения фильтрации в таблице отображаются только те строки, данные в которых соответствуют **условиям фильтрации.** Остальные – временно скрываются.

Для выбора режима фильтрации нужно установить табличный курсор в произвольную ячейку таблицы или выделить диапазон ячеек и выполнить ***Главная*** ⇒ ***Редактирование*** ⇒ ***Сортировка и фильтр*** ⇒ ***Фильтр*** ***или Данные*** ⇒ ***Сортировка и фильтр*** ⇒ ***Фильтр.***

Для установки условий фильтрации можно использовать фильтры по значению, числовые и текстовые фильтры.

Фильтрацию можно осуществлять последовательно для нескольких столбцов таблицы. Тогда каждый следующий фильтр будет применен только для тех строк таблицы, которые отображаются после применения всех предыдущих фильтров.

Для отмены фильтрации нужно выполнить ***Данные*** ⇒ ***Сортировка и фильтр*** ⇒ ***Очистить.***

Для выполнения фильтрации при условии, которое является объединением условий фильтрации двух столбцов логической операцией **ИЛИ**, используются **расширенные фильтры.** Для этого в свободных ячейках таблицы необходимо написать названия столбцов и условия фильтрации. Затем выполнить ***Данные*** ⇒ ***Сортировка и фильтр*** ⇒ ***Дополнительно*** и установить параметры фильтрации в окне **Расширенный фильтр.**

Дайте ответы на вопросы

1°. В чем заключается сортировка данных в **Excel 2007?**
2•. Какие правила сортировки данных по возрастанию в **Excel 2007?**
3•. Каким образом можно провести быструю сортировку?
4•. Когда команда **Сортировка от наименьшего к наибольшему** меняется на команду **Сортировка от А до Я?**

5•. Как осуществляется сортировка данных в таблицах по значениям нескольких столбцов?
6°. Что такое фильтрация?
7•. Какие виды фильтрации можно осуществлять для числовых данных; текстовых данных?
8•. Для чего используется расширенный фильтр?
9°. Как задать условия для расширенного фильтра?
10*. В чем отличия автоматического и расширенного фильтров?

Выполните задания

1°. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.4\образец 7.4.1xlsx**). На листе **Лист1** представлена таблица вылета самолетов из аэропортов Киева. Отсортируйте таблицу по возрастанию времени вылета самолетов. На листе **Лист2** установите режим фильтрации. Отобразите данные только о самолетах, вылетающих из аэропорта «Борисполь». Сохраните книгу в вашей папке в файле с именем **задание 7.4.1.xlsx**.

2•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.4\образец 7.4.2.xlsx**). На листе **Лист1** представлена таблица результатов участия спортсменов в соревнованиях по троеборью: фехтование (очки), бег (время), прыжки в высоту (м). Отсортируйте таблицу по значениям в трех столбцах: **Очки** (по убыванию), **Время** (по возрастанию), **Высота** (по убыванию). На листе **Лист2** установите режим фильтрации. Отобразите данные об участниках, которые набрали в фехтовании не менее 40 очков и прыгнули в высоту от 2,20 до 2,25 м. Сохраните книгу в вашей папке в файле с именем **задание 7.4.2.xlsx**.

3•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.4\образец 7.4.3.xlsx**). На листе **Лист1** представлена таблица вылета самолетов из аэропортов Киева. Отсортируйте таблицу так, чтобы даные о самолетах, вылетающих из одного аэропорта, шли подряд, данные об аэропорте назначения размещались в порядке от Я до А, а время вылета увеличивалось. На листе **Лист2** установите режим фильтрации. Отобразите данные о самолетах, вылетающих в Варшаву с 9 до 15 часов. Сохраните книгу в вашей папке в файле с именем **задание 7.4.3.xlsx**.

4•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.4\образец 7.4.4.xlsx**). На листе **Лист1** представлена таблица с данными о жестких магнитных дисках. Задайте условие расширенной фильтрации для отбора данных о ЖМД, в которых или емкость больше 500 Гбайт, или цена меньше 1000 грн. Сохраните книгу в вашей папке в файле с именем **задание 7.4.4.xlsx**.

5*. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.4\образец 7.4.5.xlsx**). На листе **Лист1** представлена таблица с данными о TFT-мониторах. Задайте условие расширенной фильтрации для отбора данных о мониторах, в которых или диагональ больше 19″, или цена меньше 3000 грн. Сохраните книгу в вашей папке в файле с именем **задание 7.4.5.xlsx**.

7.5. ПРОМЕЖУТОЧНЫЕ ИТОГИ. УСЛОВНОЕ ФОРМАТИРОВАНИЕ

1. Что такое сортировка данных в электронной таблице? Какими способами ее можно выполнить?
2. Что такое фильтрация данных в электронной таблице? Какими способами ее можно выполнить?
3. Что такое расширенная фильтрация? Приведите пример ее использования.

ПРОМЕЖУТОЧНЫЕ ИТОГИ

Часто в одном или нескольких столбцах таблицы содержатся данные, которые повторяются. Это могут быть, например, названия кружков, которые посещают ученики; названия фирм-поставщиков, товаров, пунктов назначения движения поездов и др.

На рисунке 7.31 представлена таблица с данными о продаже дисков с фильмами в нескольких магазинах.

Как мы видим, в трех столбцах этой таблицы (**Магазин**, **Жанр** и **Название**) есть данные, которые повторяются. Анализируя данные этой таблицы, можно выяснить, например, сколько экземпляров фильмов каждого жанра было продано, какой магазин продал больше дисков и на какую сумму, сколько экземпляров каждого фильма и на какую сумму было продано и т. п. Такой анализ позволяет планировать, какой из жанров имеет

	A	B	C	D	E	F	G
1	**Код**	**Магазин**	**Жанр**	**Название фильма**	**Цена, грн**	**Количество, шт.**	**Продано, грн**
2	546	Мир вокруг	Боевик	Терминатор	9	2	18
3	621	Сатурн	Фантастика	Звездные войны	13	3	39
4	876	Континент	Драма	Жестяной кубок	7	3	21
5	999	Континент	Фильм ужасов	Закат-убежище вампиров	4	3	12
6	432	Сатурн	Фантастика	Назад в прошлое	12	4	48
7	677	Стиль	Фантастика	Лангольеры	14	4	56
8	888	Континент	Фильм ужасов	Иствикские ведьмы	7	4	28
9	999	Мир вокруг	Фильм ужасов	Закат-убежище вампиров	4	5	20
10	543	Мир вокруг	Драма	Под роялем	8	6	48
11	654	Стиль	Драма	Уолл-стрит	7	6	42
12	876	Сатурн	Драма	Жестяной кубок	7	6	42
13	987	Стиль	Фильм ужасов	Армия тьмы	5	6	30
14	432	Мир вокруг	Фантастика	Назад в прошлое	12	7	84
15	543	Мир вокруг	Комедия	Близнецы	6	7	42
16	543	Континент	Комедия	Близнецы	6	7	42
17	543	Континент	Драма	Под роялем	8	7	56
18	546	Стиль	Боевик	Терминатор	9	7	63
19	654	Континент	Драма	Уолл-стрит	7	7	49
20	678	Сатурн	Комедия	Бум 1,2	6	7	42
21	765	Мир вокруг	Драма	Аптечный ковбой	5	7	35
22	987	Мир вокруг	Фильм ужасов	Армия тьмы	5	7	35
23	543	Сатурн	Драма	Под роялем	8	8	64
24	654	Сатурн	Драма	Уолл-стрит	7	8	56
25	765	Сатурн	Драма	Аптечный ковбой	5	8	40
26	888	Сатурн	Фильм ужасов	Иствикские ведьмы	7	8	56
27	987	Сатурн	Фильм ужасов	Армия тьмы	5	8	40
28	999	Сатурн	Фильм ужасов	Закат-убежище вампиров	4	8	32
29	123	Стиль	Боевик	Крепкий орешек	13	9	117
30	543	Сатурн	Комедия	Близнецы	6	9	54
31	543	Стиль	Драма	Под роялем	8	9	72

Рис. 7.31. Данные о дисках и их продаже в магазинах

	A	B	C	D	E	F
1	Магазин	Жанр	Название фильма	Цена, грн	Количество, шт.	Продано,грн
2	Континент	Боевик	Двойной удар	6	19	114
3	Мир вокруг	Боевик	Терминатор	9	2	18
4	Мир вокруг	Боевик	Двойной удар	6	10	60
5	Сатурн	Боевик	Двойной удар	6	12	72
6	Сатурн	Боевик	Крепкий орешек	13	45	585
7	Стиль	Боевик	Терминатор	9	7	63
8	Стиль	Боевик	Двойной удар	6	12	72
9	Стиль	Боевик	Крепкий орешек	13	9	117
10	Континент	Драма	Жестяной кубок	7	3	21
11	Континент	Драма	Аптечный ковбой	5	9	45
12	Континент	Драма	Уолл-стрит	7	7	49
13	Континент	Драма	Под роялем	8	7	56
14	Мир вокруг	Драма	Аптечный ковбой	5	7	35
15	Мир вокруг	Драма	Под роялем	8	6	48
16	Сатурн	Драма	Аптечный ковбой	5	8	40
17	Сатурн	Драма	Жестяной кубок	7	6	42
18	Сатурн	Драма	Уолл-стрит	7	8	56
19	Сатурн	Драма	Под роялем	8	8	64
20	Стиль	Драма	Уолл-стрит	7	6	42
21	Стиль	Драма	Жестяной кубок	7	9	63
22	Стиль	Драма	Под роялем	8	9	72
23	Континент	Комедия	Близнецы	6	7	42
24	Континент	Комедия	Бум 1,2	6	15	90
25	Континент	Комедия	Шельменко денщик	10	12	120
26	Континент	Комедия	Свадьба в Малиновке	12	15	180
27	Континент	Комедия	За двома зайцями	11	19	209
28	Мир вокруг	Комедия	Близнецы	6	7	42
29	Мир вокруг	Комедия	Шельменко денщик	10	17	170
30	Мир вокруг	Комедия	Свадьба в Малиновке	12	16	192
31	Мир вокруг	Комедия	За двома зайцями	11	31	341

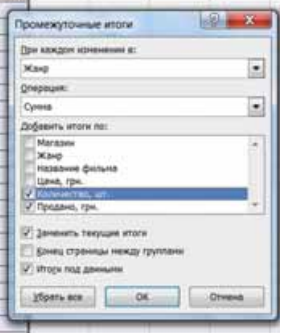


Рис. 7.32. Таблица, отсортированная по столбцу **Магазин**, и окно **Промежуточные итоги**

больший спрос, какие фильмы более популярны, какой магазин работает эффективнее и др.

Чтобы получить ответы на эти и аналогичные вопросы, следует объединить данные в группы, состоящие из повторяющихся значений (жанр, название фильма, магазин) и затем уже проводить необходимые вычисления (например, нахождение суммы, среднего, наибольшего и т. д.) уже внутри каждой группы отдельно.

Одним из способов выполнить такие операции – является **нахождение промежуточных итогов**.

Чтобы найти промежуточные итоги, следует:

1. Выделить ячейки, для значений которых будут находиться промежуточные итоги. Например, на рисунке 7.32 это диапазон **A:F**.
2. Выполнить сортировку выделенного диапазона в том столбце, по которому необходимо сгруппировать данные. Например, столбец **B** – **Жанр**.
3. Выполнить ***Данные*** ⇒ ***Структура*** ⇒ ***Промежуточные итоги***.
   Выбрать в окне **Промежуточные итоги** (рис. 7.32) в списке поля **Операция** ту операцию, которая будет обрабатывать данные (*Сумма*, *Количество*, *Среднее*, *Максимум* и т. д.). Например, на рисунке 7.32 это операция **Сумма**.
4. Установить в окне **Промежуточные итоги** в списке поля **Добавить итоги по** отметки флажков у тех столбцов, для которых будут вычислены промежуточные итоги. Например, на рисунке 7.32 это столбцы **Количество** и **Продано**.
5. Выбрать кнопку **ОК**.

На рисунке 7.33 представлен результат выполнения этих действий для таблицы, рассмотренной выше. Данные в таблице объединены в группы по

	A	B	C	D	E	F
1	Магазин	Жанр	Название фильма	Цена, грн	Количество, шт.	Продано, грн
2	Континент	Боевик	Двойной удар	6	19	114
3	Мир вокруг	Боевик	Терминатор	9	2	18
4	Мир вокруг	Боевик	Двойной удар	6	10	60
5	Сатурн	Боевик	Двойной удар	6	12	72
6	Сатурн	Боевик	Крепкий орешек	13	45	585
7	Стиль	Боевик	Терминатор	9	7	63
8	Стиль	Боевик	Двойной удар	6	12	72
9	Стиль	Боевик	Крепкий орешек	13	9	117
10		*Боевик Итог*			*116*	*1101*
24		*Драма Итог*			*93*	*633*
43		*Комедия Итог*			*267*	*2611*
44	Континент	Фантастика	Космическая полиция	6	14	84
45	Континент	Фантастика	Звездные войны	13	11	143
46	Континент	Фантастика	Синий бархат	9	16	144
47	Континент	Фантастика	Лангольеры	14	50	700
48	Мир вокруг	Фантастика	Космическая полиция	6	13	78
49	Мир вокруг	Фантастика	Назад в прошлое	12	7	84
50	Сатурн	Фантастика	Звездные войны	13	3	39
51	Сатурн	Фантастика	Назад в прошлое	12	4	48
52	Сатурн	Фантастика	Похищение Европы	13	17	221
53	Стиль	Фантастика	Лангольеры	14	4	56
54	Стиль	Фантастика	Космическая полиция	6	15	90
55	Стиль	Фантастика	Синий бархат	9	18	162
56	Стиль	Фантастика	Похищение Европы	13	19	247
57		*Фантастика Итог*			*191*	*2096*
69		*Фильм ужасов Итог*			*77*	*422*
70		Общий итог			744	6863

Рис. 7.33. Таблица с промежуточными итогами в поле **Количество** и **Продано** по значению столбца **Жанр**

значениям в столбце **Жанр** и для каждой группы создана строка **Итог.** Для столбцов **Количество** и **Продано** подсчитана *сумма* чисел в каждой группе. Первоначально каждая из групп полностью раскрыта, о чем свидетельствуют кнопки − слева от каждой из них. Например, на рисунке 7.33 для жанров *Боевик, Фантастика*. Если выбрать одну из этих кнопок, то соответствующая группа строк свернется и будет отображаться только строка **Итог**, а кнопка слева от группы примет вид +. Например, для жанров на рисунке 7.33 *Драма, Комедия*. В любой момент можно выбрать эту кнопку и снова развернуть соответствующую группу. В самом конце таблицы подведен общий итог по количеству и сумме продаж.

Над кнопками сворачивания/разворачивания групп находятся три кнопки 1, 2 и 3 быстрого сворачивания/разворачивания уровней сгруппированной таблицы: первый уровень – это вся таблица, второй – группы, третий – члены групп.

УСЛОВНОЕ ФОРМАТИРОВАНИЕ

Еще одним способом выбрать в таблице значения, которые удовлетворяют определенным условия, есть так называемое ***условное форматирование.***

Условное форматирование автоматически изменяет формат ячейки на заданный, если выполняется некоторое условие.

Например, в электронном журнале класса удобно обозначать разными цветами данные об учениках, которые имеют оценки низкого, среднего, достаточного и высокого уровней. Например, ячейки, значения которых находятся в диапазоне от 1 до 3, выделить красным цветом, оценки от 4 до 6 – зеленым, от 7 до 9 – желтым, от 10 до 12 – синим цветом.

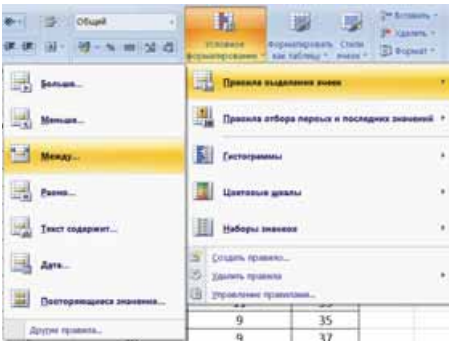


Рис. 7.34. Кнопка **Условное форматирование** и список типов выделения ячеек

Обращаем ваше внимание: в отличие от фильтрации, условное форматирование не скрывает ячейки, значения в которых не удовлетворяют заданному условию, а только определенным образом выделяет в таблице те ячейки, значения в которых удовлетворяют заданным условиям.

В **Excel 2007** для условного форматирования существует пять способов выделения ячеек, список которых открывается выбором кнопки **Условное форматирование** в группе **Стили** на вкладке **Главная** (рис. 7.34):

1. Правила выделения ячеек.
2. Правила отбора первых и последних значений.
3. Гистограммы.
4. Цветовые шкалы.
5. Наборы значков.

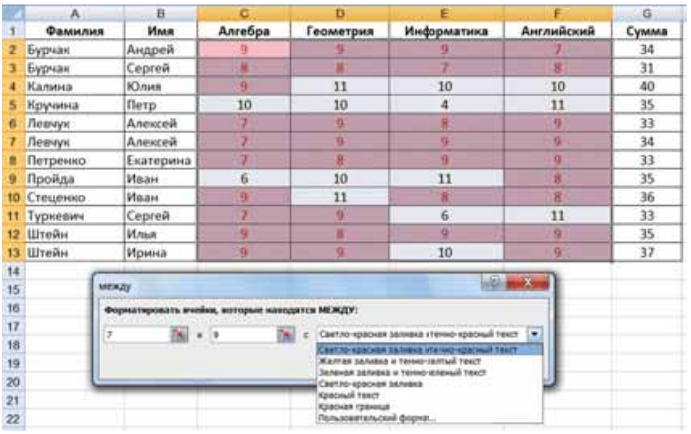
Для установки первых двух видов условного форматирования, а именно установление **правил форматирования**, следует:

1. Выделить необходимый диапазон ячеек.
2. Выполнить ***Главная*** ⇒ ***Стили*** ⇒ ***Условное форматирование***.
3. Выбрать в списках кнопки **Условное форматирование** нужный способ (рис. 7.34).
4. Указать в списке выбранного способа нужное правило форматирования.
5. Задать в открывшемся окне, условие форматирования и формат, который будет установлен, если условие будет выполняться (рис. 7.35). Формат можно выбрать из стандартных или, выбрав команду **Пользовательский формат**, установить собственный в окне **Формат ячеек**.
6. Выбрать кнопку **ОК**.

На рисунке 7.35 представлены в качестве примера окно **между**, в котором установлено условие форматирования **между 7 и 9**, со списком стандартных форматов, а также предварительный просмотр выбранного условного форматирования ячеек таблицы.

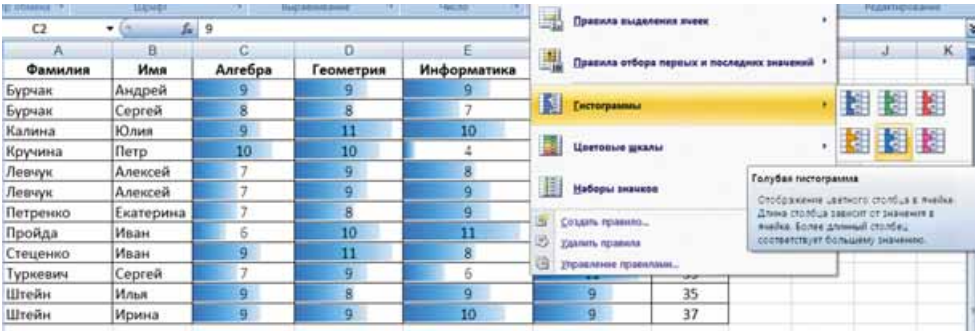
Установка условного форматирования типа **Гистограммы** приводит к вставке в ячейки выделенного диапазона гистограмм, длина цветной полосы которых пропорциональна значению в ячейке (рис. 7.36).

Установка условного форматирования типа **Цветовые шкалы** приводит к заливке в выделенном диапазоне ячеек с одинаковыми значениями одним и тем же цветом (рис. 7.37).



	A	B	C	D	E	F	G
1	Фамилия	Имя	Алгебра	Геометрия	Информатика	Английский	Сумма
2	Бурчак	Андрей	9	9	9	7	34
3	Бурчак	Сергей	8	8	7	8	31
4	Калина	Юлия	9	11	10	10	40
5	Кручина	Петр	10	10	4	11	35
6	Левчук	Алексей	7	9	8	9	33
7	Левчук	Алексей	7	9	9	9	34
8	Петренко	Екатерина	7	8	9	9	33
9	Пройда	Иван	6	10	11	8	35
10	Стеценко	Иван	9	11	8	8	36
11	Туркевич	Сергей	7	9	6	11	33
12	Штейн	Илья	9	8	9	9	35
13	Штейн	Ирина	9	9	10	9	37

Рис. 7.35. Выделенный диапазон ячеек и окно **между** с условием форматирования и форматом из стандартного списка



	A	B	C	D	E
1	Фамилия	Имя	Алгебра	Геометрия	Информатика
2	Бурчак	Андрей	9	9	9
3	Бурчак	Сергей	8	8	7
4	Калина	Юлия	9	11	10
5	Кручина	Петр	10	10	4
6	Левчук	Алексей	7	9	8
7	Левчук	Алексей	7	9	9
8	Петренко	Екатерина	7	8	9
9	Пройда	Иван	6	10	11
10	Стеценко	Иван	9	11	8
11	Туркевич	Сергей	7	9	6
12	Штейн	Илья	9	8	9
13	Штейн	Ирина	9	9	10

Рис. 7.36. Условное форматирование типа **Гистограммы**



	A	B	C	D	E
1	Фамилия	Имя	Алгебра	Геометрия	Информатика
2	Бурчак	Андрей	9	9	9
3	Бурчак	Сергей	8	8	7
4	Калина	Юлия	9	11	10
5	Кручина	Петр	10	10	4
6	Левчук	Алексей	7	9	8
7	Левчук	Алексей	7	9	9
8	Петренко	Екатерина	7	8	9
9	Пройда	Иван	6	10	11
10	Стеценко	Иван	9	11	8
11	Туркевич	Сергей	7	9	6
12	Штейн	Илья	9	8	9
13	Штейн	Ирина	9	9	10

Рис. 7.37. Условное форматирование типа **Цветовые шкалы**

Можно также выбрать вид условного форматирования из списка **Наборы значков.** Во время такого форматирования в ячейках выделенного диапазона появится значок из выбранного набора. Появление конкретной

пиктограммы из набора в ячейке означает, что значение в этой ячейке удовлетворяет условию, установленному для каждого значка набора.

Обращаем ваше внимание: к одному диапазону может быть применено несколько правил условного форматирования, выполненных последовательно.

Для удаления условного форматирования следует выполнить ***Условное форматирование ⇒ Удалить правила ⇒ Удалить правила из выделенных ячеек (или Удалить правила со всего листа).***

Самое важное в этом пункте

Промежуточные итоги позволяют вычислить некоторые операции (количество, сумма, максимальное, среднее) для групп ячеек, состоящих из повторяющихся значений. При этом сначала данные объединяются в группы, затем производятся необходимые вычисления внутри каждой группы отдельно.

Кнопка **Промежуточные итоги**, которая устанавливает данный режим вычислений, находится на вкладке **Данные** в группе **Структура**.

Условное форматирование автоматически изменяет формат ячейки на заданный, если выполняется некоторое условие. Существует пять способов выделения ячеек: ***Правила выделения ячеек, Правила отбора первых и последних значений, Гистограммы, Цветовые шкалы, Наборы значков***. Их список открывается выбором кнопки **Условное форматирование** в группе **Стили** на вкладке **Главная**.

Дайте ответы на вопросы

1°. Что такое промежуточные итоги? Когда их целесообразно создавать?
2•. Как вычислить промежуточные итоги в **Excel 2007**?
3•. Какие функции можно вычислять в промежуточных итогах?
4° Что такое условное форматирование? Для чего оно выполняется?
5°. Какие есть способы условного форматирования?
6•. Как выполнить условное форматирование?
7°. Как очистить условное форматирование на всем листе?

Выполните задания

1°. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.5\образец 7.5.1.xlsx**). На листе **Лист1** представлена таблица со значениями свойств винчестеров. Вычислите промежуточные итоги по значениям в столбце **Фирма**. Сохраните книгу в вашей папке в файле с именем **задание 7.5.1.xlsx**.

2•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.5\образец 7.5.2.xlsx**). На листе **Лист1** представлена таблица со значениями свойств мониторов. Вычислите промежуточные итоги по значениям в столбце **Диагональ**. Сохраните книгу в вашей папке в файле с именем **задание 7.5.2.xlsx**.

3•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.5\образец 7.5.3.xlsx**). На листе **Лист1** представлена таблица с дан-

ными о TFT-мониторах. Закрасьте в розовый цвет все ячейки, в которых диагональ монитора меньше 21″. Сохраните книгу в вашей папке в файле с именем **задание 7.5.3.xlsx**.

4•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.5\образец 7.5.4.xlsx**). На листе **Лист1** представлена таблица с данными о жестких магнитных дисках. Установите синий цвет ячеек, в которых емкость кэш-памяти больше 8 Мбайт. Сохраните книгу в вашей папке в файле с именем **задание 7.5.4.xlsx**.

5•. Откройте указанный учителем файл (например, **Глава 7\Пункт 7.5\образец 7.5.5.xlsx**). На листе **Лист1** представлена таблица с данными о температуре за последний месяц в г. Киеве. Установите условное форматирование **Гистограммы** (*синяя гистограмма*) для всех числовых данных таблицы. Сохраните книгу в вашей папке в файле с именем **задание 7.5.5.xlsx**.



6•. Найдите в Интернете данные о количестве населения в вашем областном центре за последние 5 лет. На листе **Лист1** создайте электронную таблицу по этим данным. Установите условное форматирование с использованием **Наборов значков** (*4 сигнала светофора*) для данных таблицы. Сохраните книгу в вашей папке в файле с именем **задание 7.5.6.xlsx**.

ПРАКТИЧЕСКАЯ РАБОТА № 16

«Сортировка данных в таблицах. Автоматические и расширенные фильтры. Условное форматирование»

Внимание! *Работая с компьютером, соблюдайте правила безопасности и санитарно-гигиенические нормы.*

1. Откройте указанный учителем файл (например, **Глава 7\Практическая 16\образец пр16.xlsx**).
2. На листе **Лист1** выполните сортировку таблицы по значениям в столбце **Фамилия** *по убыванию*.
3. На листе **Лист2** выполните сортировку таблицы по значениям в нескольких столбцах в указанном порядке: **Фамилия** (*по возрастанию*), **Имя** (*по возрастанию*), **Год рождения** (*по убыванию*).
4. На листе **Лист3** отобразите данные об учениках, рост которых *выше 1 м 80 см*.
5. На листе **Лист4** отобразите данные об учениках, которые посещают кружки *рисования* или *пения*.
6. На листе **Лист5** вычислите промежуточные итоги по значениям в столбце **Вес учеников**, которые определяют средний и максимальный вес.
7. На листе **Лист6** установите условное форматирование, чтобы ячейки, в которых вес учащихся больше 80 кг, закрашивались красным цветом, а ячейки, в которых вес учащихся меньше 75 кг, – синим.
8. Сохраните книгу в вашей папке в файле с именем **практическая 16.xlsx**.

Глава 8. Решение компетентностных задач, выполнение учебных проектов

В этой главе вы узнаете:

- об этапах решения задач из разных отраслей человеческой деятельности с использованием информационно-коммуникационных технологий
- о стратегии поиска информационных материалов
- о действиях на каждом из этапов решения компетентностных задач с использованием информационно-коммуникационных технологий
- о выполнении учебных проектов по информатике

8.1. ЭТАПЫ РЕШЕНИЯ КОМПЕТЕНТНОСТНЫХ ЗАДАЧ

1. Что такое компетентностная задача?
2. Каков алгоритм решения задач из разных областей человеческой деятельности?
3. Какие средства могут быть использованы для представления решения компетентностных задач?

ЭТАПЫ РЕШЕНИЯ КОМПЕТЕНТНОСТНЫХ ЗАДАЧ

В седьмом классе вы начали решать задачи, которые называют компетентностными. Напомним, что **компетентностными** называют задачи из разных областей деятельности человека, требующие от человека умения использовать полученные знания на практике. Их смысл заключается в решении некоторой жизненной проблемы с применением знаний, умений и навыков, полученных на уроках по различным предметам. Значительная часть таких задач не ограничивается предметной областью одного учебного предмета, а является межпредметной.

Вы уже знаете этапы решения компетентностных задач (рис. 8.1).

Рассмотрим подробнее названные этапы.

Выполнить содержательный анализ формулировки задачи означает выделить в формулировке детали задания и явно указанные исходные данные, определить потребность в других данных, необходимых для решения задачи, наметить, что будет считаться конечным результатом.



Рис. 8.1. Этапы решения компетентностных задач

Осуществить поиск информационных материалов, выполнив две задачи:

- Разработать **стратегию поиска** информационных материалов, определив:
  - цели и задания поиска;
  - источники, в которых будет осуществляться поиск;
  - ключевые слова;
  - критерии оценивания найденных материалов.
- Осуществить поиск информационных материалов, в том числе в интернет-ресурсах, в соответствии с разработанной стратегией, сопоставить результаты поиска с целью, оценить в соответствии с критериями.

Стратегия (древнегреч. στρατηγία – искусство полководца) – общий, недетализированный план некоторой деятельности, охватывающий продолжительный период; способ достижения сложной цели.

Построить информационную модель задачи означает составить план и описание решения задачи, определить последовательность действий, которые приведут к выполнению поставленной задачи.

Выбрать средства обработки данных в зависимости от поставленной задачи и имеющихся материалов. Средствами может быть текстовый процессор, графический редактор, редактор презентаций, табличный процессор, учебная среда выполнения алгоритмов, редактор мультимедийных данных и др. Во многих задачах может быть использовано несколько средств.

Обработка данных предусматривает применение выбранных программных средств для реализации плана решения задачи, описанного в информационной модели задачи. На этом этапе выполняется сравнение и сопоставление сведений, полученных из разных источников, исключение неподходящих и несущественных сведений, структурирование данных, получение новых данных на основе известных, формулировка выводов.

Представить результаты решения задачи в соответствии с аудиторией означает разработать итоговый документ, в котором отображается ход и результаты решения задачи, и представить его аудитории. При этом существенным является учет назначения итогового документа, краткое и логичное изложение обобщенных данных, обоснование выводов.

ПРИМЕР РЕШЕНИЯ КОМПЕТЕНТНОСТНОЙ ЗАДАЧИ

Вспомним этапы решения компетентностных задач на примере следующей задачи: *Ваш класс собирается во время учебной практики поехать на однодневную экскурсию в один из областных центров Украины. Цель экскурсии – ознакомиться с памятными местами, связанными с событиями, которые изучались в курсе истории Украины в 8 классе. Чтобы выбрать город для поездки, договорились выяснить, какие памятные места изученного периода истории Украины можно увидеть в каждом городе, и разработать маршруты пешеходных экскурсий. Маршрут может быть дополнен посещением достопримечательностей, связанных с выдающимися личностями, известными из курса истории Украины или украинской литературы. Каждый учащийся класса собирает сведения об одном из городов. После обсуждения в классе найденных сведений будет выбран город проведения экскурсии. Вам поручили собрать сведения о городе Полтаве. Для удобства выполнения задания вам предоставлен файл с картой Полтавы* (рис. 8.2).

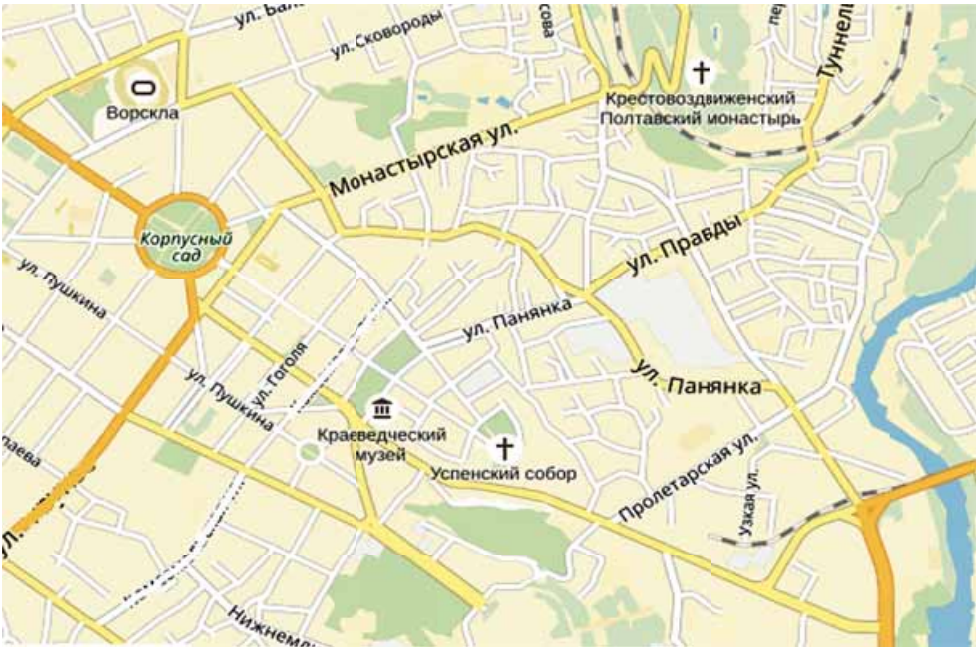


Рис. 8.2. Карта Полтавы

1. Во время **содержательного анализа формулировки задачи** следует выяснить:

а) Что необходимо получить как решение задачи?

- Проложить на карте маршрут экскурсии по городу Полтаве.
- Обеспечить длительность экскурсии на проложенном маршруте, подходящую для однодневной поездки.
- Представить исторические памятки на маршруте.

б) Какие данные, необходимые для решения задачи, есть в условии?

- Места, выбранные для экскурсии, должны касаться событий, которые изучались в курсе истории Украины в 8 классе (*Украина в XVI–XVIII вв.*), и исторических лиц, известных из истории Украины или украинской литературы.
- Длительность пешеходной экскурсии должна быть в пределах одного дня.

в) Какие данные необходимо найти или вспомнить из ранее изученного для решения задачи?

- Какие события из истории Украины XVI–XVIII вв. и исторические личности связаны с Полтавой? (*Полтавская битва 1709 г. – по данным из учебника истории Украины для 8 класса; Г. Сковорода, Т. Шевченко, М. Чурай, И. Котляревский – по данным учебников истории Украины и украинской литературы*).
- Какие достопримечательности в современной Полтаве связаны с историческими событиями и личностями?
- Какие достопримечательности, связанные с историческими событиями и личностями, можно посетить в течение однодневной пешеходной экскурсии?

2. **Поиск информационных материалов** необходимо выполнить для нахождения сведений о памятных местах и их расположении на территории Полтавы.

а) Определим **стратегию поиска**:

- **Цель поиска:** получить описание и данные о расположении в Полтаве памятных мест, связанных с историческими событиями в Украине XVI–XVIII вв. и историческими личностями.
- **Задание поиска:** найти тексты с описанием достопримечательностей, их адреса, фотографии, карты с обозначенными местами расположения достопримечательностей.
- **Источники поиска** списка исторических достопримечательностей: печатные энциклопедии, справочники, путеводители, Интернет. Сведения об исторических достопримечательностях могут быть получены как с официальных сайтов города, его музеев, галерей, так и с сайтов энциклопедий, туристических фирм, личных сайтов и блогов пользователей, посещавших Полтаву.
- **Ключевые слова** для поиска сведений об исторических достопримечательностях:
  - Полтава XVI–XVIII вв.;
  - исторические места Полтавы;
  - исторические личности Полтавы;
  - литературная Полтавщина;
  - экскурсия в Полтаву;
  - памятники Полтавской битвы;
  - музей Полтавской битвы.
- **Критерии оценивания** найденных материалов:
  - материалы должны быть актуальными, не устаревшими;
  - должна существовать возможность проверить найденные данные по данным из других источников;
  - изображения должны быть качественными.

б) В результате поиска и оценки найденных ресурсов были получены следующие сведения:

- В Полтаве с событиями XVI–XVIII вв. связаны следующие достопримечательности:
  - Государственный историко-культурный заповедник «Поле Полтавской битвы» (*battle-poltava.org*);
  - Музей истории Полтавской битвы (*tourism.poltava.ua/muzei/Muzej_istorii_Poltavskoi_bitvi*);
  - исторические памятники на территории города Полтавы за пределами заповедника «Поле Полтавской битвы» (*histpol.pl.ua/ru/poltava-istoricheskie-ocherki/poltavskaya-bitva-i-ejo-pamyatniki*).
- С историческими личностями на территории города Полтавы связаны такие достопримечательности:
  - мемориальный музей-усадьба И. Котляревского;
  - памятники Т. Шевченко, М. Чурай, И. Котляревскому.

3. После анализа найденных материалов можно составить словесную информационную модель задачи:

Большинство исторических памятников Полтавы, связанных с событиями XVI–XVIII вв., расположены на территории Государствен-

ного историко-культурного заповедника «Поле Полтавской битвы» и рядом с Музеем истории Полтавской битвы на северной окраине города. Это десять гранитных обелисков на месте бывших редутов, памятник шведам от русских, шведам от шведов, Сампсониевская церковь и др. Экскурсию по заповеднику можно заказать в Музее истории Полтавской битвы.

В центре города может быть проведена пешеходная экскурсия, маршрут которой начинается у монумента Славы, проходит мимо Памятника погибшим украинским казакам, Спасской церкви, памятника Т. Шевченко, музея-усадьбы И. Котляревского и завершается у Белой альтанки, откуда открывается панорама на Крестовоздвиженский монастырь.

Расстояние от первой до последней точки маршрута составляет около 2 км, что соответствует условию пешеходной однодневной экскурсии.

4. Для представления результатов решения задачи следует **определить, какие средства для этого могут быть использованы.**

   Проложить маршрут пешеходной экскурсии на карте можно в среде графического редактора. Для представления памятных мест на маршруте может быть использован редактор презентаций, текстовый процессор или редактор видео.
5. **Обработка данных**, то есть создание итоговых документов, будет выполняться в зависимости от выбранных программ.
6. Один из вариантов **представления результатов** решения задачи приведен на рисунках 8.3 и 8.4.



Рис. 8.3. Пример презентации с описанием исторических памятников Полтавы

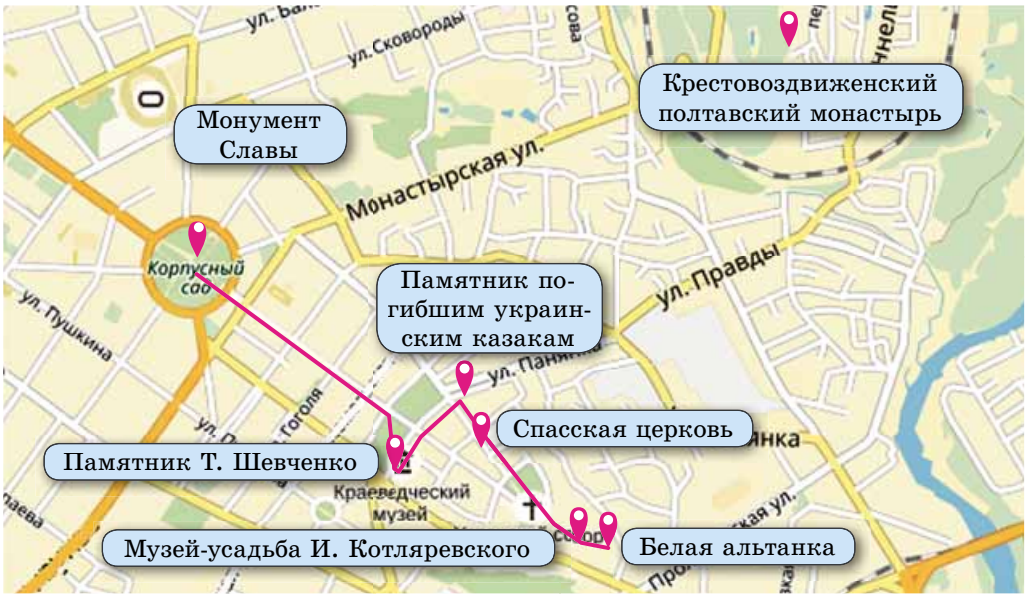


Рис. 8.4. Пример карты с маршрутом по памятным местам Полтавы

Самое важное в этом пункте

При решении задач следует придерживаться следующих этапов:

1. Содержательный анализ формулировки задачи.
2. Поиск необходимых материалов и сведений.
3. Построение информационной модели задачи.
4. Выбор средств обработки данных.
5. Обработка данных.
6. Представление результатов решения задачи.

Разработка **стратегии поиска** информационных материалов заключается в определении:

- цели и заданий поиска;
- источников, в которых будет осуществляться поиск;
- ключевых слов;
- критериев оценивания найденных материалов.

Дайте ответы на вопросы

1•. Что такое компетентностная задача?
2•. Какие этапы решения компетентностных задач?
3•. Какие действия выполняют на этапе содержательного анализа формулировки задачи?
4•. На основании чего определяют, какие материалы и сведения необходимы для решения задачи?
5•. В чем заключается разработка стратегии поиска информационных материалов?

6°. В каких формах может быть представлена информационная модель задачи?
7•. Какие действия выполняют на этапе обработки данных?
8•. От чего зависит вид представления результатов решения задачи?
9°. Какие средства обработки данных с использованием компьютеров вы изучили в восьмом классе?

Выполните задания

1•. Постройте схему маршрута экскурсии и описание исторических памятников Полтавы по образцу на рисунке 8.3 (текстовые и фотоматериалы находятся в папке **Глава 8\Пункт 8.1\Задание 8.1.1**) с использованием:
а°) графического редактора, например **Paint**, и редактора презентаций, например **PowerPoint**;
б•) текстового процессора, например **Word**, для изображения схемы маршрута экскурсии и редактора видео для создания видеосюжета об исторических достопримечательностях;
в*) среды исполнения алгоритмов, например **Scratch**, для создания анимации движения указателя вдоль маршрута экскурсии и текстового процессора, например **Word**, для описания достопримечательностей.

2•. Постройте схему маршрута однодневной пешеходной экскурсии и составьте описание памятных мест, связанных с событиями из истории Украины XVI–XVIII вв. и выдающимися историческими личностями в вашем областном центре. Расчет длины маршрута выполните в табличном процессоре. Другие средства обработки данных выберите самостоятельно.

3*. Постройте схему маршрута однодневной пешеходной экскурсии и составьте описание памятных мест, связанных с событиями из истории Украины XVI–XVIII вв. и выдающимися историческими личностями в любом областном центре Украины по вашему выбору. Рассчитайте стоимость экскурсии, включая проезд, питание, заказ платных экскурсий. Расчеты стоимости выполните в табличном процессоре, другие средства обработки данных выберите самостоятельно.

4•. Найдите сведения и создайте памятку для уроков трудового обучения: в текстовом процессоре – как выбирать режим стирки одежды в зависимости от обозначений, приведенных на одежде; в редакторе презентаций – как выбирать режим глажки одежды в зависимости от ткани, из которой изготовлена одежда, и обозначений, приведенных на одежде.

5•. Найдите сведения и создайте презентацию для урока географии, в которой расскажите об особенностях расселения жителей Украины по разным областям. Диаграммы расселения выполните в табличном процессоре.

6•. Найдите рецепты и фотографии блюд, которые готовят в разных странах к празднику Пасхи. Создайте в текстовом процессоре подборку праздничных блюд для использования на уроках трудового обучения.

Составьте программу, например в среде **Lazarus**, для вычисления стоимости традиционной украинской пасхальной корзины.

7°. Найдите рецепты пасхального кулича. Создайте видеоурок по приготовлению пасхи по одному из рецептов. В табличном процессоре вычислите стоимость продуктов для приготовления пасхи по выбранному рецепту.

8*. Вы планируете классом пойти в поход в лес. Найдите сведения о способах ориентирования в лесу без компаса и приемы выживания в сложных погодных условиях. Создайте в текстовом процессоре памятку с найденными сведениями. В табличном процессоре выполните расчеты количества питьевой воды, необходимой для похода в зависимости от его продолжительности и количества участников, а также стоимости покупки нужного количества воды в зависимости от емкости упаковок.

9*. Ваша семья решила сделать ремонт в вашей комнате. Вам предложили выбрать обои, рассчитать количество рулонов, необходимых для ремонта, их стоимость и представить родителям ваше видение интерьера комнаты и полученные расчеты. Средства для решения задачи и представления результатов родителям выберите самостоятельно.

8.2. УЧЕБНЫЕ ПРОЕКТЫ

1. На уроках по каким предметам вы выполняли учебные проекты?
2. Каковы этапы реализации проектов?
3. В каких формах могут быть представлены результаты выполнения проектов?

ЭТАПЫ РЕАЛИЗАЦИИ ПРОЕКТОВ

Как вы знаете, проектная деятельность учащихся заключается в самостоятельном поиске и обработке сведений, необходимых для решения некоторой проблемной задачи, подготовке отчета о проделанной работе и защите результатов поисковой работы.

Вспомним этапы реализации проектов:

- анализ формулировки задачи (определение цели и заданий проекта);
- создание информационной модели (разработка плана выполнения проекта);
- поиск необходимых материалов и сведений;
- выбор средств обработки данных;
- обработка материалов, анализ результатов;
- выбор средств представления результатов учебного проекта;
- оформление результатов проекта;
- защита проекта.

Над выполнением заданий проекта можно работать в одиночку, но часто учебные проекты учащиеся выполняют группами.

Для создания информационных моделей в ходе проекта, поиска и обработки материалов, подготовки отчета по результатам работы, общения участников могут использоваться различные информационные технологии.

Результаты, полученные различными группами учащихся, удобно размещать на страницах сайта или блога в Интернете.

Учебные проекты вы выполняли при изучении различных школьных предметов. Проекты могут касаться одного учебного предмета или нескольких. В этом году вам предлагается выполнить индивидуальные и групповые учебные проекты по исследованию предметной области учебного курса «Информатика».

УЧЕБНЫЙ ПРОЕКТ ПО ИНФОРМАТИКЕ

Рассмотрим этапы работы над проектом на примере проекта *«Алгоритмы вокруг нас»*. Проект касается раздела 6 *«Алгоритмы работы с объектами и величинами»*.

Цель проекта: работая в группах, найти примеры алгоритмов в собственном окружении и создать компьютерные модели отдельных исполнителей алгоритмов.

Одна из групп может рассмотреть тему *«Перекресток алгоритмов»*. Для этой группы можно определить следующие **задания**:

- определить виды перекрестков, встречающихся на дорогах, и объекты на перекрестках, которые являются исполнителями алгоритмов;
- описать алгоритмы, которые выполняет каждый из исполнителей;
- выбрать одного из исполнителей и его алгоритм, разработать компьютерную модель реализации выбранного алгоритма;
- оформить результаты работы в проекте.

На основе заданий разработаем **план выполнения проекта**. В плане конкретизируем действия, связанные с выполнением каждого задания (рис. 8.5).



Рис. 8.5. План выполнения проекта

Поиск необходимых сведений может выполняться различными методами: наблюдение, опрос, поиск в Интернете и печатных источниках. В нашем проекте результатом поиска будут сведения (описания, изображения, схемы) о видах перекрестков и участниках дорожного движения на перекрестках, алгоритмы их действий.

Для обмена найденными материалами между участниками проекта и руководителем может использоваться электронная почта.

Результатом **обработки найденных сведений** может стать схема или таблица, в которой структурированы все найденные материалы. Например, в текстовом процессоре может быть создана такая таблица (табл. 8.1).

Таблица 8.1

Перекресток алгоритмов

Перекресток / ***Участники движения***	***Перекресток с регулируемым переходом***			***Перекресток с нерегулируемым переходом***
	Движение регулируется светофором	*Движение регулируется семафором со шлагбаумом*	*Движение регулируется светофором и семафором со шлагбаумом*	
Пешеход	Алгоритм с ветвлением	Алгоритм с ветвлением	Алгоритм с ветвлением	Алгоритм с ветвлением
Водитель автотранспорта	Алгоритм с ветвлением	Алгоритм с ветвлением	Алгоритм с ветвлением	Алгоритм с ветвлением
Водитель железнодорожного транспорта	—	Алгоритм с ветвлением	Алгоритм с ветвлением	—
Светофор	Алгоритм с повторением	—	Алгоритм с повторением	—
Семафор	—	Алгоритм с ветвлением и повторением	Алгоритм с ветвлением и повторением	—
Шлагбаум	—	Алгоритм с ветвлением	Алгоритм с ветвлением	—

Таблица дает основания считать, что на разных перекрестках участники дорожного движения чаще всего выполняют алгоритмы с ветвлением. Это можно объяснить требованиями Правил дорожного движения, которые гарантируют безопасность для каждого участника движения.

Каждый участник проекта может выбрать отдельного исполнителя алгоритмов на перекрестке, описать и смоделировать алгоритм его работы. Рассмотрим подробнее действия исполнителя *светофор*.

Алгоритм действия одного из видов светофоров – это алгоритм с повторением, который начинает выполняться после включения:

1. Включить лампу красного цвета на 30 секунд.
2. Выключить лампу красного цвета.

3. Включить лампу желтого цвета на 15 секунд.
4. Выключить лампу желтого цвета.
5. Включить лампу зеленого цвета на 30 секунд.
6. Выключить лампу зеленого цвета.
7. Включить лампу желтого цвета на 15 секунд.
8. Выключить лампу желтого цвета.
9. Повторять действия 1–8, пока светофор включен.

Блок-схема этого алгоритма приведена на рисунке 8.6.

Компьютерную модель светофора создадим в среде **Lazarus**. Текст программы и окно исполнения проекта приведены на рисунке 8.7.

Для **представления результатов** работы в проекте можно разработать презентацию или текстовый документ, разместить полученные материалы на сайте или блоге или выбрать другие средства. В презентации важно отобразить весь ход вашей работы: поставленные задачи, методы поиска и обработки данных, которые вы использовали, полученные результаты и выводы. Пример презентации с результатами приведен на рисунке 8.8.

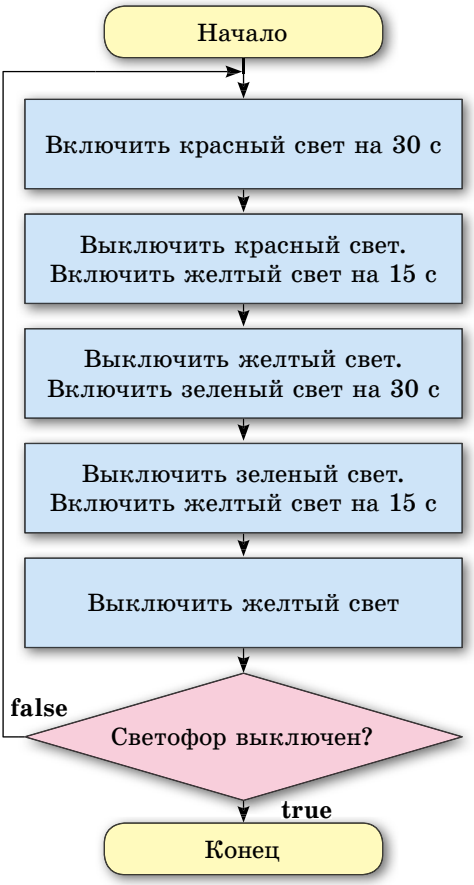


Рис. 8.6. Алгоритм работы светофора

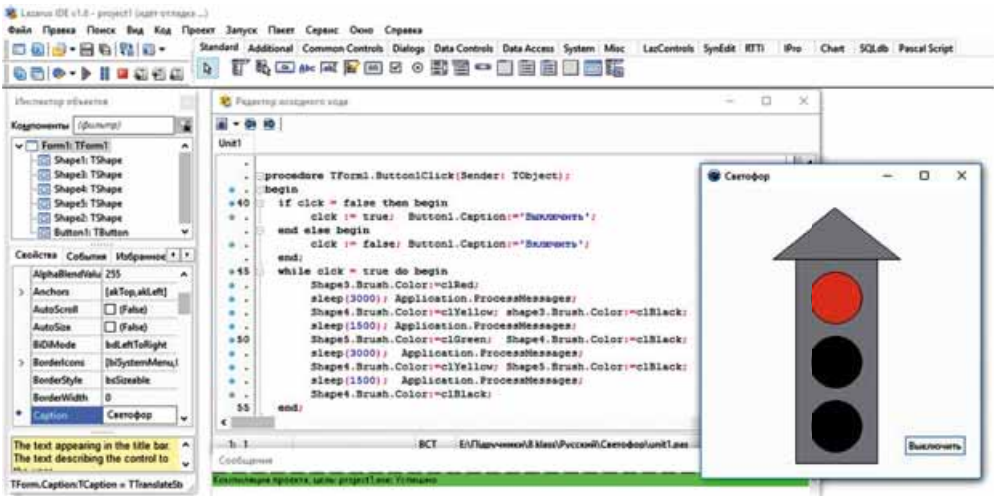


Рис. 8.7. Текст программы и окно исполнения проекта моделирования работы светофора

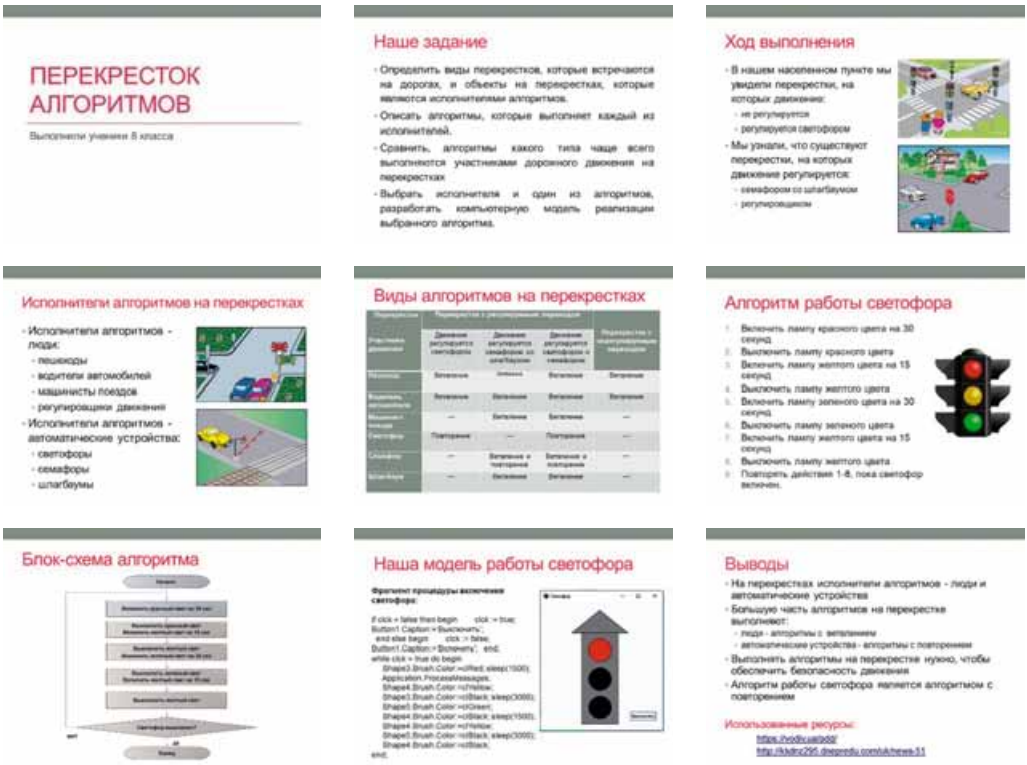


Рис. 8.8. Пример презентации с результатами работы в проекте

Завершающим этапом работы в проекте является его **защита**. Во время защиты участники проекта отчитываются о результатах реализации проекта. Планом выступления на защите будет презентация, которую вы создали. Важно не только подать описание проекта и ход его выполнения, но и отразить участие каждого из членов группы в проекте. Таким образом, в выступлении должны быть названы:

- название и цели проекта;
- средства, методы и пути, которые были выбраны для реализации проекта;
- трудности и проблемы, с которыми встретились участники проекта в ходе его реализации;
- результаты проекта;
- выводы, сделанные на основе результатов проекта;
- насколько удалось достичь цели проекта.

Дайте ответы на вопросы

1•. Каковы этапы реализации проектов?

2•. Для чего составляют план проекта?

3°. В каких источниках и какими методами можно выполнять сбор сведений для проекта?

4°. Что может быть результатом поиска материалов и сведений для проекта?

5°. В каких формах могут быть представлены результаты выполнения проекта?

6*. Что влияет на отбор компьютерных программ для обработки материалов проекта?

7•. Что обычно отображается в презентации по результатам участия в проекте?

Выполните задания

Разработайте план учебного проекта, выберите один из объектов исследования в соответствии с целью, средства обработки данных и форму представления результатов и выполните проект по информатике:

1. Тема: **Гордимся украинским**. Цель: *исследовать историю развития компьютерной техники в Украине и определить, какие достижения в этой области имеет Украина и украинцы*.
2. Тема: **Суд над компьютером**. Цель: *определить, в чем заключается вредное влияние компьютера на человека, и подготовить рекомендации для уменьшения опасности*.
3. Тема: **Мой напарник – компьютер**. Цель: *проанализировать виды и характеристики современных компьютеров, представленных на рынке вашего региона, и определить характеристики компьютеров, необходимые для выполнения заданий в различных отраслях человеческой деятельности*.
4. Тема: **Профессия – специалист в отрасли информационных технологий**. Цель: *составить справочник профессий, которые можно получить в области информационных технологий, выяснить, какими навыками должны овладеть представители этих профессий*.
5. Тема: **Веб 2.0 в школе**. Цель: *выяснить суть понятия Веб 2.0, выполнить классификацию сервисов, относящихся к Веб 2.0, и привести примеры сервисов, которые можно использовать в обучении*.
6. Тема: **Информационные технологии под микроскопом**. Цель: *обобщить знания об информационных технологиях обработки данных различных видов (графических, текстовых, числовых, мультимедийных), составить схемы со сведениями об информационных технологиях, указав основные объекты обработки, области применения, типы программных средств, историю разработки*.

СЛОВАРИК

А

Абсолютная ссылка – ссылка на ячейку, которая не модифицируется при копировании формулы, с. 230.
Архиваторы – программы, используемые для выполнения операций над архивами данных, с. 63.
Архивация – процесс создания копий данных с использованием специальных программ, которые могут использовать сжатие данных, с. 62.
Архивный файл (архив) – файл, который содержит в сжатом или не сжатом состоянии файлы и папки, с. 62.
Архитектура компьютера – модель, описывающая взаимодействие устройств и программ для обеспечения информационных процессов в компьютере, с. 25.

Б

Байт – последовательность из восьми битов, с. 18.
Бит – цифра 0 или 1 в двоичном коде сообщения, с. 17.

В

Видеоредакторы – программы для редактирования видеоданных, с. 138.
Видеостудии – программы, которые используются в профессиональных целях для редактирования видеоданных, с. 138.
Внешняя память – составляющая памяти компьютера, предназначенная для длительного хранения данных. В ее состав входят устройства на магнитных и оптических дисках, накопители, использующие флэш-память и т. п., с. 27.
Внутренняя память – составляющая памяти компьютера, предназначенная для обеспечения работы процессора. Состоит внутренняя память из оперативной, постоянной и кэш-памяти, с. 26.

Г

Граббер (англ. *grabber* – тот, кто захватывает, хапуга, грабитель) – см. ***рекордер***, с. 133.
График – диаграмма, которая используется для отображения некоторого процесса во времени, динамики изменения данных, не зависящих друг от друга, с. 243.
Графический планшет – устройство ввода графических данных. Используется свойство рабочей поверхности планшета реагировать на прикосновение стилуса. Пользователь создает рисунок, прикасаясь стилусом к поверхности графического планшета, с. 33.

Д

Двоичное кодирование – кодирование сообщений с использованием сигналов только двух видов, с. 17.
Двоичный код сообщения – сообщение, полученное в результате двоичного кодирования исходного сообщения, с. 17.
Декодирование сообщения – процесс получения исходного сообщения из закодированного, с. 8.
Длина двоичного кода сообщения (объем данных) – количество байтов в двоичном коде этого сообщения, с. 18.

И

Инсталляция – операция установки ПО на компьютере, с. 54.

К

Кириллица – алфавит для группы славянских языков, к которой относятся украинский, русский, белорусский, болгарский и другие языки, с. 11.

Клавиатура – устройство ввода, которое предназначено для ввода символьных данных и команд, с. 32.

Клип (англ. *clip* – сжимать, обрезать) – небольшой по размеру (продолжительности) кино- или видеофильм, с. 139.

Кодеки – программы, содержащие алгоритмы кодирования и декодирования мультимедийных данных, с. 132.

Кодирование сообщения – процесс замены одной последовательности сигналов, которой представлено сообщение, другой последовательностью сигналов, с. 7.

Колонтитулы – служебные сообщения, которые размещаются на полях страницы и повторяются на других страницах документа, с. 109.

Команда неполного ветвления в языке программирования Object Pascal имеет следующий вид (с. 201):

```
If <логическое выражение>
Then begin
        <последовательность команд>
     end;
```

Команда полного ветвления в языке программирования Object Pascal имеет следующий вид (с. 201):

```
If <логическое выражение>
Then begin
        <последовательность команд 1>
     end
Else begin
        <последовательность команд 2>
     end;
```

Команда цикла со счетчиком в языке программирования Object Pascal имеет следующий вид (с. 213):

```
for <имя переменной> := <выражение1> to <выражение2> do
begin
  <команды тела цикла>
end;
```

Команда цикла с предусловием в языке программирования Object Pascal имеет следующий вид (с. 217):

```
While <логическое выражение>
Then begin
        <команды тела цикла>
     end;
```

Комментарий – текст, который игнорируется компилятором и включается в текст проекта с целью облегчения его понимания и поиска логических ошибок. Комментарий следует заключить в фигурные скобки {}, или в круглые скобки со звездочками (* *), или отделить от команд двумя наклонными чертами //, с. 189.

Коммерческая лицензия – лицензия на программное обеспечение предполагает оплату пользователем стоимости использования данной программы на одном или нескольких указанных в лицензии компьютерах. Различают «коробочную», OEM и корпоративную версии коммерческой лицензии, с. 52.

Компьютерная программа – алгоритм обработки данных, записанный на специальном языке и предназначенный для исполнения компьютером. В процессе работы программа обрабатывает данные, с. 161.

Конвертор – программа для изменения форматов файлов, с. 135.

Кратные единицы измерения длины двоичного кода сообщения:

1 Кбайт (килобайт) = 2^{10} байт = 1024 байт
1 Мбайт (мегабайт) = 2^{10} Кбайт = 2^{20} байт = 1 048 576 байт
1 Гбайт (гигабайт) = 2^{10} Мбайт = 2^{20} Кбайт = 2^{30} байт
1 Тбайт (терабайт) = 2^{10} Гбайт = 2^{20} Мбайт = 2^{30} Кбайт = 2^{40} байт, с. 18.

КОИ-8 – таблица кодов символов, которая включает без изменений все коды таблицы ASCII, а также содержит коды букв кириллицы, с. 11.

КОРЕНЬ – математическая функция для вычисления арифметического квадратного корня из числа, с. 237.

Л

Линейчатая диаграмма – диаграмма используется для отображения нескольких рядов данных, не зависящих друг от друга, с. 242.

Линия – графический объект, который характеризуется свойствами: форма, толщина, тип, шаблон, цвет, с. 91.

Лицензия свободного использования программ, или **freeware** (англ. *freeware* – свободный товар), – лицензия предусматривает свободное использование программ без выплат вознаграждения автору, но не предусматривает возможности внесения изменений в программу, с. 53.

Лицензия с открытым кодом, или ***free*** **(**англ. *free* – свободный) ***software,*** или ***libre*** (исп. *libre* – свободный) ***software***, – лицензия на свободное программное обеспечение, предусматривающее не только бесплатное использование программ, но и право на их модификацию, внесение изменений в программу, с. 53.

Логическая функция (ЕСЛИ, И, ИЛИ, НЕ) – функция, результат которой равен **Истина** или **Ложь**, с. 237.

Логические выражения – математические выражения, содержащие знаки сравнений, с. 237.

М

Маркеры графических объектов – специальные отметки в виде маленьких кружков, квадратиков, ромбиков, расположенных вокруг объекта и предназначенных для изменения объекта, с. 94.

Монитор – основное устройство для вывода данных в персональных компьютерах. Их разделяют по способу создания изображения на ***мониторы на жидких кристаллах,*** или **LCD**-мониторы (англ. ***L****iquid* ***C****rystal* ***D****isplay* – жидкокристаллический дисплей), ***плазменные***, ***OLED*** (англ. ***O****rganic* ***L****ight* ***E****mitting* ***D****iode* – органический светодиод), на ***электронных чернилах*** – **e-ink** (англ. ***e****lectronic* ***ink*** – электронные чернила) и т. п., с. 34.

Музыкальные редакторы – программы для редактирования звуковых (музыкальных) данных, с. 138.

Музыкальные студии – программы, которые используются в профессиональных целях для редактирования звуковых (музыкальных) данных, с. 138.

О

Обработка данных – процесс получения новых данных из имеющихся, с. 6.

ОКРУГЛ – математическая функция для вычисления округления числа, с. 237.

Относительная ссылка – ссылка на ячейку, которая модифицируется при копировании формулы, с. 230.

П

Переменная логического типа – переменная, которая может принимать одно из двух значений **true** или **false**. Логический тип переменных обозначается **boolean,** с. 196.

Плоттер – устройство вывода данных на бумагу, пленку или другие материалы, имеющие размер А2 и больше, с. 35.
Подкаст (англ. ***ipod*** – медиаплеер корпорации Apple; *broad* ***cas****ting* – эфирное вещание) – медиафайл, распространяемый с использованием Интернета и, как правило, содержит мнения авторов по различным вопросам: политическим, экономическим, техническим, литературным, музыкальным и т. п., с. 150.
Подкастер – тот, кто создает подкасты, с. 150.
Предметный (алфавитный) указатель – алфавитный список терминов, использованных в документе, с указанием номеров страниц, где они встречаются, с. 118.
Прикладное программное обеспечение – совокупность программ, предназначенных для реализации конкретных задач обработки данных, которые пользователь решает в ходе своей деятельности, с. 51.
Принтер – устройство вывода данных на бумагу, пленку или другие материалы, которые имеют размер, как правило, не более чем А3, с. 35.
Пробная, или **trial** (англ. *trial* – испытание), **лицензия** – вид лицензии, которая предоставляется для пробного использования программы в течение определенного времени (обычно около месяца) или на определенное количество запусков программ, с. 52.
Программное обеспечение (ПО) – совокупность программ для компьютеров, с. 51.
Промежуточные итоги – способ выполнения вычислений в электронной таблице с предыдущим объединением повторяющихся данных в группы, с. 259.
Процессор – устройство, которое обеспечивает обработку данных в компьютере. Он содержит **устройство управления**, которое обеспечивает выполнение команд компьютерной программы, и **арифметико-логическое устройство**, которое осуществляет операции над данными, с. 25.
ПИ – математическая функция для вычисления приближенного значения числа π = **3,14159265358979**, с. 237.

Р

Рекордер (англ. *record* – запись) – программа для записи (захвата) аудио- и видеоданных, с. 133.

С

Свойства формы:

1. **Name** (англ. *name* – имя) – его значение определяет имя формы как объекта, с. 157.
2. **Caption** (англ. *caption* – заголовок) – его значение определяет текст в строке заголовка формы, с. 157.
3. **Color** (англ. *color* – цвет) – его значение определяет цвет фона формы, с. 157.
4. **Height** (англ. *height* – высота) – его значение определяет высоту окна в пикселях (**пиксель** – это точка графического изображения на экране), с. 157.
5. **Width** (англ. *width* – ширина) – его значение определяет ширину окна в пикселях, с. 157.
6. **Top** (англ. *top* – верх) – его значение определяет расстояние (в пикселях) от верхней границы окна до верхней границы экрана, 157.
7. **Left** (англ. *left* – левый) – его значение определяет расстояние (в пикселях) от левой границы окна до левой границы экрана, с. 157.
8. **Enabled** (англ. *enabled* – доступный) – его значение определяет, доступен ли объект для операций с ним. Это свойство может принимать только одно из двух значений: **True** (англ. *true* – истина) – объект доступен или **False** (англ. *false* – ложь) – объект недоступен, с. 171.

9. **Visible** (англ. *visible* – видимый) – его значение определяет, видимый ли данный объект в окне. Это свойство может также принимать только одно из двух значений: **True** – объект видимый или **False** – объект невидимый, с. 171.
10. **Font** (англ. *font* – шрифт) – его значение определяет значение свойств шрифта, которым будет выводиться текст на объекте. Это свойство является комплексным, потому что состоит из нескольких свойств шрифта: **Color**, **Height**, **Name**, **Size** и др., с. 171.

Сенсорный экран – устройство ввода и вывода данных. Для ввода данных используется свойство поверхности экрана реагировать на прикосновение пальца или стилуса, с. 33.

Системы программирования – совокупность программ, предназначенных для создания новых программ с использованием различных языков программирования, с. 51.

Системное программное обеспечение – совокупность программ, предназначенных для управления работой устройств компьютера и обменом данными между ними, диагностики и устранения неполадок в работе компьютера, автоматизации процесса обработки данных, организации обмена данными между пользователем и компьютером. В состав системного программного обеспечения входят операционные системы и различные утилиты, с. 51.

Сжатие данных – процесс перекодировки данных с целью уменьшения размеров файлов, с. 60.

Сканер – устройство для ввода графических данных. По принципу действия выделяют планшетные, с протяжным механизмом, ручные и скан-камеры, с. 33.

Служебное программное обеспечение – программы, предназначенные для диагностирования аппаратной и программной составляющих компьютера, расширения возможностей ОС, с. 51.

Смешанная ссылка – ссылка на ячейку, в которой при копировании формулы модифицируется или номер столбца, или номер строки, с. 231.

Содержание документа – перечень названий структурных частей документа с указанием номеров соответствующих страниц, с. 118.

Сортировка – изменение порядка расположения данных в строках или столбцах таблицы по некоторым признакам, с. 248.

Списки – особый вид форматирования абзацев текстового документа, который используют для перечня объектов, описания порядка действий и т. п., с. 73.

Ссылка – специальная запись в документе, которая указывает на какой-либо объект этого или другого документа, с. 118.

Стиль – набор значений свойств определенного типа объектов, который имеет собственное имя, с. 111.

СТЕПЕНЬ – математическая функция для вычисления числа, возведенного в степень, с. 237.

Стратегия поиска информационных материалов – определение цели и заданий поиска, источников, в которых будет осуществляться поиск ключевых слов, критериев оценивания найденных материалов, с. 267.

Структура документа – иерархическая схема размещения составных частей документа, с. 108.

Т

Таблица кодов символов – таблица, где каждому символу, который может быть использован в текстовом сообщении, соответствует некоторое число, с. 10.

Таблица символов – служебная программа операционной системы **Windows**, предназначенная для просмотра, поиска и вставки в документы символов из разных

шрифтов, представленных с использованием различных таблиц кодов символов, с. 13.

Табуляция – средство, которое дает возможность размещать объекты в строке текстового документа в строго определенных местах (позициях табуляции), с. 76.

Тачпад (англ. *touch* – прикосновение, *pad* – площадка, подушечка) – устройство ввода, которое используется в ноутбуках и нетбуках для управления работой компьютера, с. 32.

ТВ-тюнер – устройство, обеспечивающее просмотр телевизионных передач на компьютере, с. 34.

Тестирование проекта – процесс поиска логических ошибок в проекте с использованием тестовых наборов входных данных, с. 188.

Тестовый набор входных данных – набор значений входных данных, для которых можно заранее определить правильный результат исполнения проекта, после чего сравнить его с реальным результатом исполнения проекта для этих значений входных данных, с. 188.

Тип переменной определяет:

- какие значения может принимать эта переменная;
- какие операции с ней можно выполнять;
- какой объем оперативной памяти будет выделен для хранения значения этой переменной, с. 184.

Точка остановки – строка в проекте, перед исполнением которой исполнение проекта приостанавливается, с. 190.

Точечная диаграмма – диаграмма, которая используется для демонстрации взаимосвязи между двумя переменными, когда значение одной величины зависит от значения другой, с. 244.

У

Условное форматирование – вид форматирования данных, который автоматически изменяет формат ячейки электронной таблицы на заданный, если выполняется некоторое условие, с. 260.

Ф

Фигура – графический примитив, который образуется замкнутыми линиями, с. 91.

Фильтрация – выбор значений в ячейках электронной таблицы, соответствующих определенным условиям, с. 251.

Форма – объект, на котором можно разместить различные компоненты, например, кнопки, поля, метки, меню, полосы прокрутки и др. В ходе исполнения проекта форма открывается как окно приложения, на котором отображаются все объекты, расположенные на форме, с. 157.

Ш

Шаблон – отформатированный определенным образом документ-заготовка, который сохранен в отдельном файле и используется как образец для создания новых документов определенного типа, с. 121.

Ю

Юникод (англ. *Unicode* – **уни**фицированное **код**ирование) – таблица кодов символов, которая состоит из 17 наборов по 65 536 значений кодов в каждом и дает возможность закодировать 1 114 112 разных символов, с. 11.

A

ABS – математическая функция для вычисления модуля числа, с. 237.
ASCII – таблица кодов символов, содержит коды 128 символов (букв английского алфавита, цифр, знаков препинания, специальных символов), с. 10.

F

FLAC (англ. ***F**ree **L**ossless **A**udio **C**odec* – свободный аудиокодек без потерь) – формат файлов, который использует сжатие аудиоданных без потерь и распространяется по лицензии свободного программного обеспечения, с. 131.
Flash видео (расширение имени файла **flv**) – формат файлов, используемых обычно для распространения видео в Интернете. Характеризуется высокой степенью сжатия данных, с. 132.

M

MIDI (англ. ***M**usical **I**nstrument **D**igital **I**nterface* – цифровой интерфейс музыкальных инструментов) – формат файлов, содержащих команды для воспроизведения звука специальным устройством или программой – синтезатором, с. 131.
MP3, точнее **MPEG-1/2/2.5La*y*er 3** (англ. ***M**otion **P**icture **E**xperts **G**roup* – экспертная группа по вопросам движущегося изображения, ***Layer 3*** – третий уровень), – формат аудиофайлов с частичной потерей данных. Используется для передачи звуковых данных в глобальных сетях, в файлообменных системах и т. п., с. 131.
MP4, или PEG-4 Part 14, – файлы этого формата обычно используются для цифрового телевидения, хранения видеофильмов и т. п., с. 131.
QuickTime (англ. ***Q**uick **T**ime* – быстрое время) – формат файлов, основанный на технологии хранения и воспроизведения видео- и звуковых данных, разработанной корпорацией Apple, с. 132.

S

ShowMessage (‘<текст>’); – команда для открытия окна сообщений и вывода в нем текстового сообщения, с. 167.

W

WAV, или **WAVE** (англ. ***wav**eform audio format* – аудиоформат типа *волна*), – формат звуковых файлов, используется как основной для несжатых аудиоданных в операционных системах семейства **Windows**, с. 131.
Windows-1251 – таблица кодов символов, включает без изменений все коды таблицы **ASCII**, а также содержит коды букв кириллицы. Является стандартом для кодирования букв кириллицы в операционных системах **Windows,** с. 11.
WMA/WMV (англ. ***W**indows **M**edia **A**udio/**V**ideo*) – формат файлов для хранения и воспроизведения аудио-, видеоданных, разработанный корпорацией Microsoft и стандартно используется в ОС Windows, с. 132.

Y

YouTube (англ. *You Tube* – твоя труба (трубка)) – сервис для размещения видеофайлов в Интернете, с. 147.