

# ИНФОРМАТИКА

Учебное пособие для 7 класса  
общеобразовательных учреждений  
с русским языком обучения

Под редакцией Г. А. Зaborовского

*Допущено  
Министерством образования  
Республики Беларусь*

Минск  
«Народная асвета»  
2009

Правообладатель Народная асвета

УДК 004(075.3=161.1)  
ББК 32.81я721  
И74

Авторы:

Г. А. Заборовский (главы 2, 4, 5), А. А. Козинский (глава 2),  
А. Е. Пупцев (главы 1, 2, 4, 5), Н. В. Якунина (глава 3)

Рецензенты:

кафедра информатики и информационных технологий Витебского  
государственного университета имени П. М. Машерова  
(зав. каф., канд. физ.-мат. наук, доцент А. И. Бочкин);  
учитель информатики высшей категории  
СШ № 11 г. Слуцка Минской области В. А. Жук

И74 **Информатика** : учеб. пособие для 7-го кл. общеобразо-  
ват. учреждений с рус. яз. обучения / Г. А. Заборовский  
[и др.] ; подред. Г. А. Заборовского. — Минск : Нар. асвета,  
2009. — 159 с. : ил.

ISBN 978-985-03-1184-9.

УДК 004(075.3=161.1)  
ББК 32.81я721

ISBN 978-985-03-1184-9

© Оформление. УП «Народная  
асвета», 2009

Правообладатель Народная асвета

## ОТ АВТОРОВ

*Уважаемые школьники!*

В седьмом классе Вы продолжите изучение информатики. Умение применять компьютер и современное программное обеспечение так же важно для человека, как умение читать и писать.

В первой главе пособия рассказано о том, что такое информация. Вы узнаете о видах информации, информационных процессах, познакомитесь с примерами носителей информации.

Вторая глава посвящена изучению устройств компьютера и программного обеспечения. Вы сможете узнать о процессоре, материнской плате, внешних устройствах, видах памяти компьютера. Здесь рассказывается о назначении операционной системы, способах хранения информации. Полученные умения помогут вести диалог с операционной системой, выполнять основные операции над папками, файлами, ярлыками. В этой главе Вы узнаете о назначении локальной компьютерной сети и ее ресурсах.

В третьей главе продолжено изложение основ алгоритмизации и программирования. С помощью полученных знаний Вы сможете записывать алгоритмы, содержащие переменные, вводить данные и выводить результаты выполнения арифметических операций.

Четвертая глава знакомит с компьютерными презентациями. Умения использовать презентации помогут Вам делать свои доклады и выступления интересными, красочными и наглядными.

В последней, пятой главе рассказывается о назначении глобальной компьютерной сети Интернет, Всемирной информационной паутине, а также о работе

с программами браузерами, предназначенными для просмотра страниц сети Интернет.

Главы учебного пособия разбиты на параграфы, а параграфы — на пункты. После некоторых из них предлагаются вопросы и упражнения, которые позволяют Вам закрепить свои знания о компьютере и практические умения работы на нем.

Желаем успехов в изучении информатики и информационных технологий. Хорошие знания помогут Вам выбрать профессию и приносить пользу Родине.

# Глава 1

## ИНФОРМАЦИЯ И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ

### § 1. Виды информации

Информацию мы начинаем получать с раннего детства. Получая информацию, маленькие дети учатся понимать, говорить, играть. Когда мы становимся старше и идем в школу, мы начинаем учиться писать, читать и считать, обрабатывая различную информацию.

Трудно даже на секунду представить себе, что человек лишился привычной для него информации: новостей по телевидению либо радио; сведений из книг, газет, журналов и сети Интернет; разговоров по телефону, бесед с друзьями и родственниками.

Под информацией мы будем понимать сведения о предметах, событиях, явлениях и процессах окружающего мира.

Моряки для передачи сведений (информации) используют различные комбинации флаглов, связисты — азбуку Морзе, военные — сигнальные ракеты, люди с нарушением слуха — язык жестов.

Сведения о людях хранятся в документах, удостоверяющих их личность: паспортах, водительских удостоверениях. Отпечатки пальцев человека несут информацию, с помощью которой его можно идентифицировать.

Человек получает информацию с помощью органов зрения, слуха, вкуса, обоняния и осязания.

Глаза (орган зрения) позволяют человеку воспринимать форму, размер и цвет предметов, отслеживать перемещение объектов. Художники, например,

различают десятки различных оттенков цвета, что помогает им при создании картин.

Уши (орган слуха) дают возможность человеку воспринимать звуки: речь, музыку, пение птиц, шум моря, телефонные звонки. Известно, например, что великий скрипач Никколо Паганини слышал на расстоянии более 10 метров переговаривающихся шепотом людей.

Информацию о запахах человек получает с помощью носа (органа обоняния). Человек может различать запахи, например, полыни, сирени, яблока и др. Многие запахи усиливают восприятие человеком окружающего мира. Например, запах бергамотного масла обостряет зрительные ощущения, а запах герани — слух.

С помощью языка (органа вкуса) человек может определить вкус пищи: горькая или кислая, сладкая или соленая. Найдены, например, древнейшие записи о том, что повара Древнего Рима по вкусу рыбы определяли, в каком месте реки Тибр она была поймана.

На ощупь с помощью органов осязания человек может определить характер поверхности предмета: гладкая или шершавая, мокрая или сухая. Люди в темноте не могут определить цвет предметов. Но если один из них сделан из резины, а другой — из стекла, то человек в темноте может это различить.

Получение информации важно не только для людей, но и для растений и животных.

Растения получают различную информацию из окружающей среды. Например, в зависимости от температуры и влажности воздуха и почвы, протяженности светового дня ускоряется или замедляется их рост.

Среди многих звуков в лесу: шума деревьев, шелеста травы, криков птиц — животные безошибочно выделяют нужные им сигналы и определяют их источники.

Собаки спасательных служб благодаря обонянию находят людей под снегом. Они чувствуют их запах, но хуже воспринимают запахи трав и цветов.

В повседневной жизни мы встречаемся с различными формами представления информации. По форме представления информацию можно разделить на следующие виды: *текстовая, графическая, звуковая, комбинированная*.

**Текстовая** (алфавитно-цифровая) информация записывается с помощью букв, цифр, знаков.

Запись информации с помощью текста выполняется на разных языках (сентябрь — слово, записанное на русском языке, верасень — на белорусском, september — на английском). С помощью цифр и арифметических знаков записываются математические выражения (15 — число, записанное арабскими цифрами, XV — римскими,  $125 : 5 - 10 = 15$  — в виде арифметического выражения).

**Графическая** (визуальная) информация может быть представлена в виде фотографий, картин, рисунков, которые не содержат текста.

К звуковой информации (аудиоинформации) относятся музыка, речь, шум.

**Комбинированная** (смешанная) информация представляет собой сочетание нескольких ее видов. Наглядным примером комбинированной информации является сочетание текстовой и графической информации на географических картах.

Примером смешанной информации является также видеоинформация (от лат. *video* — смотрю, вижу). Под видеоинформацией, прежде всего, понимают одновременное воспроизведение визуальной и аудиоинформации. Примером использования видеоинформации является просмотр видеороликов и фильмов на компьютере.

- ? 1. Что такое информация?
2. Приведите примеры:
- текстовой информации;
  - графической (визуальной) информации;
  - звуковой информации (аудиоинформации);
  - комбинированной (смешанной) информации.

### *Упражнение*

Определите, к какому виду по форме представления относится следующая информация:

a)		б)		в)	$4^2 - 3^2 = 7$
г)	Химические знаки 	д)	Пры неабходнасці доўгай працы на камп'ютэры трэба рабіць перапынкі, каб захаваць свой зрок у добрым стане.	е)	

### *§ 2. Информационные процессы. Носители информации*

Объекты живой и неживой природы обладают способностью передавать, получать, хранить и обрабатывать информацию.

Процессы передачи, хранения, поиска и обработки информации называют **информационными процессами**.

Процесс *передачи* информации всегда предполагает наличие *источника* и *получателя (приемника)* информации.

Источниками информации являются различные объекты и процессы: люди, животные, растения, планеты, технические устройства, научные эксперименты, явления природы. Например, густой дым в лесу может быть источником информации о том, что начался пожар.

Специальные датчики, расположенные в сейсмоопасном районе, фиксируют изменение обстановки окружающей среды и являются для человека источником информации о возможном землетрясении.

Источником информации является движение пчел, которые таким образом передают соседкам по улью, куда необходимо лететь за нектаром.

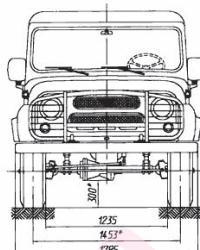
В Индии встречаются целые заросли мимозы стыдливой (*mimosa pudica*). Когда первые капли дождя попадают на одно из растений, все остальные сворачивают свои листья, спасаясь от тропического дождя, так как сигнал о его начале передается другим.

Люди, животные, растения могут быть и приемниками информации. Например, приемниками информации являются ученики, слушающие объяснение учителя.

Каждый человек в зависимости от ситуации может выступать в роли источника или приемника. Ученик, работающий над сочинением, является источником информации, а учитель, проверяющий это сочинение, — приемником информации.

Важнейшим информационным процессом является *хранение* информации. Для хранения информации используются *носители информации*.

**Носителем информации** может быть любой материальный объект, на котором размещается и хранится информация.

		
Древний рисунок на скале	Глобус	Чертеж

		
Фотография	Журналы	Лазерный диск

Рис. 1.1

Носителями информации являются: человек, животные, растения, планеты, камни, книги, газеты, рисунки, киноленты и многое другое (рис. 1.1).

С начала своего существования человек сохранял важные сведения лишь в собственной биологической памяти (мозге). Затем для хранения информации он начал использовать зарубки на деревьях, чтобы не заблудиться во время охоты, узелки или камешки для счета предметов.

Потребность запоминать все больший объем информации привела к возникновению письменности. Для записи информации стали использовать папирус, глиняные дощечки, бересту.

С появлением книгопечатания носителями информации стали книги, затем газеты и журналы.

Носителями информации могут быть объекты не живой природы. Например, с помощью спутников и планетоходов человек смог определить многие свой-

ства планет Солнечной системы, состав веществ, которые на них находятся, состояние окружающей среды.

Носителем информации в растительном мире может служить ствол дерева, информация в котором хранится в виде колец. По ним можно определить не только возраст дерева, но и то, какими были отдельные годы — засушливыми или дождливыми.

Если человеку необходимо найти интересующие его сведения, то он организует *поиск* информации. Искать нужную информацию можно различными способами: читать энциклопедии, словари, журналы и газеты, просматривать телевизионные передачи, обращаться к библиотечным каталогам и компьютерной сети Интернет.

*Обработка* информации подразумевает изменение ее формы представления или содержания.

Например, человек, который увидел и запомнил какой-либо предмет, может впоследствии описать его устно или письменно. Он обработал информацию, представленную зрительно, а затем воспроизвел ее в виде звуков или в виде текста.

В процессе обработки информации может изменяться ее содержание или появляться новая информация. Например, введенный на компьютере текст можно отредактировать: дополнить или сократить, заменить отдельные слова, а также отформатировать: изменить шрифт, цвет, начертание слов. Эти операции относятся к процессу обработки информации.

- ?
- 1. Что называют информационными процессами?
- 2. Приведите примеры процессов:
  - а) передачи информации;
  - б) хранения информации;
  - в) поиска информации;
  - г) обработки информации.
- 3. Приведите примеры носителей информации.

## Глава 2

# АППАРАТНОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

### §3. Устройства компьютера

Совокупность всех устройств компьютера называют его **аппаратным обеспечением**. Состав этих устройств (конфигурация компьютера) может изме-

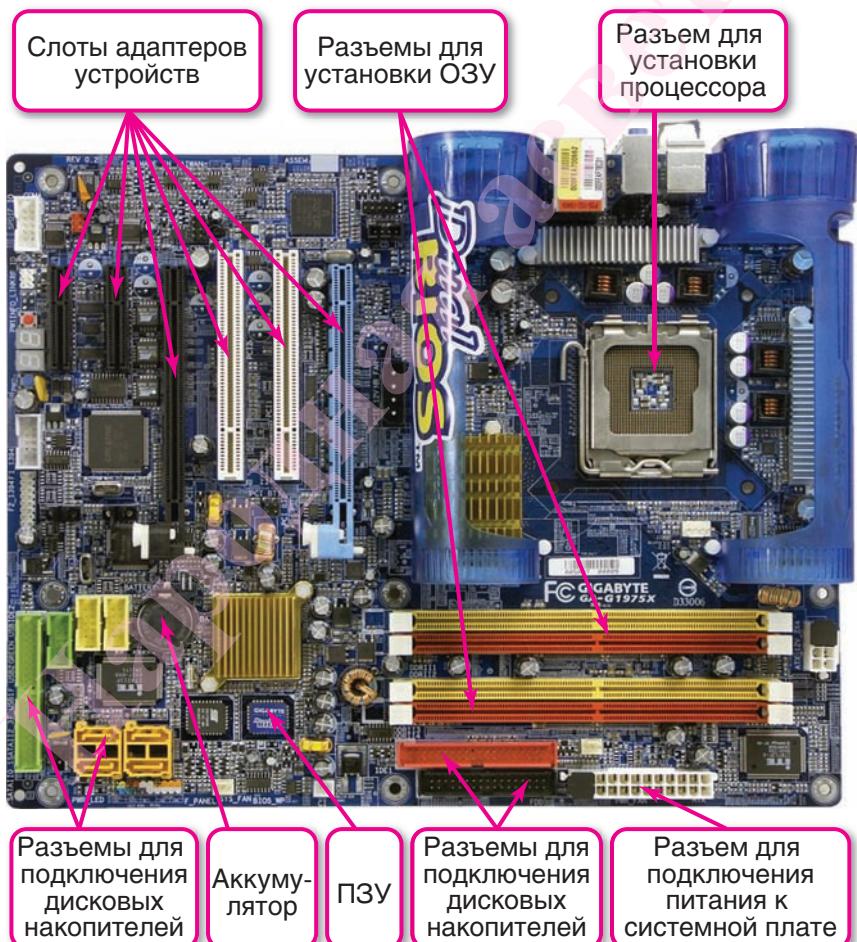


Рис. 2.1

няться в зависимости от решаемых задач. Базовая конфигурация содержит следующие функциональные блоки: системный блок, монитор, клавиатуру, мышь.

**Системный блок** представляет собой корпус с размещенными внутри него системной платой, блоком питания и другими устройствами, которые называют *внутренними*: например, дисководами для магнитных и оптических дисков. Не входящие в системный блок устройства называют *внешними*.

Основной компонент системного блока — **системная (или материнская) плата** (рис. 2.1). На ней размещаются процессор, устройства памяти, набор микросхем, управляющих работой устройств, разъемы для подключения устройств.

**Процессор** — важнейшее устройство компьютера, его мозг, предназначен для выполнения вычислений и исполнения программ (рис. 2.2).

Основные характеристики процессора, определяющие его быстродействие: тактовая частота, разрядность, модель.

**Тактовая частота** — это число тактов, которые процессор выполняет за одну секунду. Различные операции могут занимать один или несколько тактов. Тактовая частота измеряется в герцах (Гц). У современных процессоров тактовая частота достигает нескольких миллиардов герц. Поэтому ее измеряют в производных единицах: один



Рис. 2.2

мегагерц (МГц) равен миллиону герц, один гигагерц (ГГц) — это миллиард герц.

Процессоры выполняют операции не над десятичными числами, а над двоичными. Двоичные числа записываются цифрами 0 и 1, которым соответствуют единицы информации, называемые битами. **Разрядность процессора** — это количество двоичных цифр (битов), которые одновременно обрабатывает процессор. Большинство современных процессоров имеет разрядность 32 бит или 64 бит.

Производительность процессора может быть повышена за счет одновременного выполнения нескольких последовательностей операций (потоков). Процессоры, способные одновременно обрабатывать несколько потоков, являются **многоядерными**. Так, процессоры Intel Core 2 Duo и Athlon64x2 являются двухъядерными, а Intel Core Quad и AMD Quad-Core Phenom — четырехъядерными.

Другое важнейшее устройство компьютера — **память**. Для хранения информации используются различные виды памяти (запоминающие устройства).

**Оперативная память (ОЗУ** — оперативное запоминающее устройство) предназначена для хранения данных и программ во время их выполнения. Обмен данными между процессором и оперативной памятью выполняется за очень короткие промежутки времени. Информация в оперативную память поступает после включения компьютера, а при его выключении не сохраняется.

ОЗУ представляет собой набор микросхем, которые собраны в модули (рис. 2.3). Для установки модулей на материнской плате имеются соответствующие разъемы.



Рис. 2.3

Для постоянного хранения информации даже при выключенном компьютере предназначено **постоянное запоминающее устройство (ПЗУ)**. Хранимые в ПЗУ данные и программы обеспечивают запуск компьютера и работу с клавиатурой, монитором и другими устройствами.

Первоначально ПЗУ записывались однократно и были доступны только для чтения. Совершенствование технологий привело к появлению перезаписываемой энергонезависимой **флэш-памяти**, которая в настоящее время широко используется не только в ПЗУ, но и во внешних устройствах памяти.

К компьютеру можно подключать различные внутренние и внешние устройства для ввода и вывода, хранения и обработки информации.

Основным устройством долговременного хранения информации является **накопитель на жестких магнитных дисках (винчестер)**. Он представляет собой набор вращающихся дисков с магнитным покрытием, у поверхности которых расположены головки для записи — чтения.

Для переноса данных используют съемные носители: гибкие магнитные диски (дискеты), оптические диски (CD и DVD), флэш-память и др.

Максимальный объем информации, который может быть записан на носитель, называют его **емкостью**. За единицу измерения объема информации

принят байт. Емкость носителей измеряют в производных единицах:

- 1 Кбайт (килобайт) = 1024 байт;
- 1 Мбайт (мегабайт) = 1024 Кбайт;
- 1 Гбайт (гигабайт) = 1024 Мбайт;
- 1 Тбайт (терабайт) = 1024 Гбайт.

Например, информационный объем дискеты размером 3,5 дюйма равен 1,44 Мбайт, объем DVD-диска — 4,7 Гбайт, а объем винчестеров достигает нескольких сотен гигабайт.

Необходимый объем ОЗУ зависит от решаемых задач и используемых программ. Так, для работы офисных приложений Windows XP желательно не менее 512 Мбайт, а для Windows Vista — более 1 Гбайт.

При работе компьютера все его устройства взаимодействуют между собой. Они обмениваются данными друг с другом при помощи электрических сигналов. Для передачи данных от одного устройства к другому используются группы проводников на материнской плате — **шины**. Внешние устройства (например, клавиатура, мышь, принтер) подключаются через специальные разъемы (порты) (рис. 2.4). Ши-

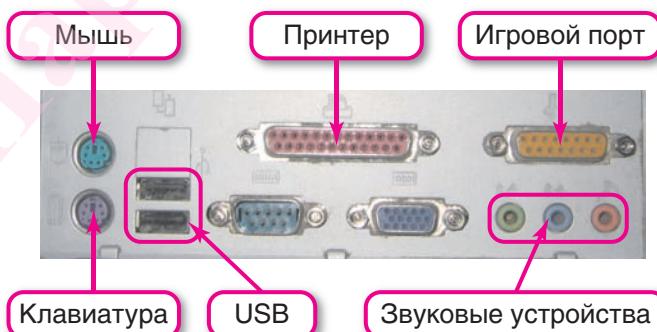


Рис. 2.4

рокое распространение в последнее время получил порт USB (Universal Series Bus — универсальная последовательная шина), допускающий «горячее» (т. е. без выключения компьютера) подключение разнообразных устройств.

Для подключения некоторых устройств требуются специальные платы — **адаптеры**. Для этого на материнской плате есть разъемы, которые называют **слотами**. Например, для подключения монитора необходим видеоадаптер (видеокарта) (рис. 2.5). Следует заметить, что многие современные материнские платы имеют встроенные видеокарты, звуковые адаптеры, модемы, сетевые адаптеры.

Некоторые внешние устройства являются обязательными компонентами компьютера: монитор, клавиатура, мышь. Другие подключаются по мере необходимости, например принтер, сканер, графи-

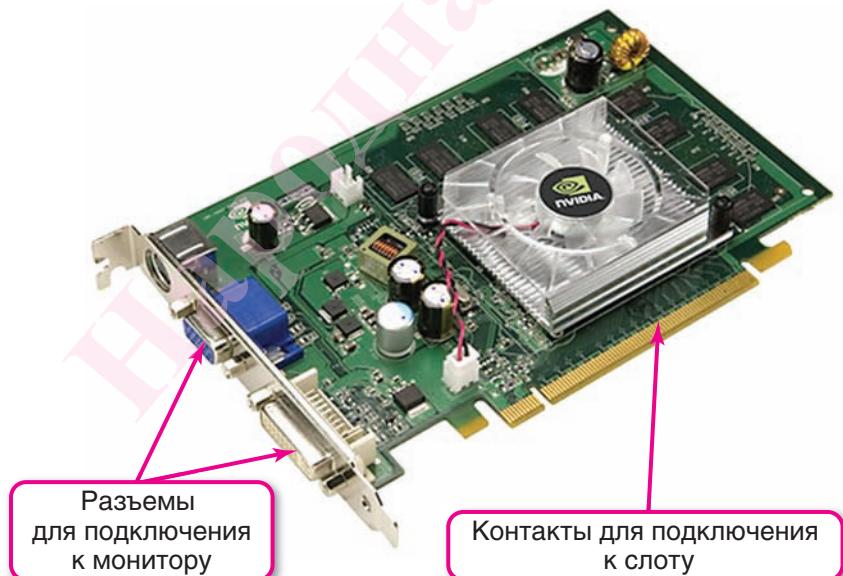


Рис. 2.5

ческий планшет (дигитайзер), микрофон, наушники или звуковые колонки, мультимедийный проектор и др.

**Монитор** предназначен для визуального представления информации. В настоящее время используются мониторы двух типов: на основе электронно-лучевых трубок и плоские жидкокристаллические, которые получают все большее распространение. Важнейший параметр монитора — размер диагонали экрана. Сегодня наиболее распространены мониторы с диагональю 15—19 дюймов (1 дюйм = 2,54 см).

В качестве устройств вывода информации на бумагу наибольшее распространение получили матричные, лазерные и струйные **принтеры**. Принцип действия матричного принтера основан на переносе на бумагу краски с ленты в результате удара иголок печатающей головки. В лазерном принтере краска (тонер) переносится на бумагу участками поверхности светочувствительного барабана, получившими световой импульс лазера. В струйном принтере капельки краски распыляются печатающей головкой.



Рис. 2.6

В последнее время широко используются устройства ввода аудиовизуальной информации: сканеры, цифровые фотоаппараты, веб-камеры, микрофоны. **Сканер** (рис. 2.6) предназначен для ввода неподвижных изображений. **Веб-камера** (рис. 2.7) позволяет

вводить видео и при наличии соответствующих программ обеспечивать видеосвязь или видеонаблюдение.

Для ввода звуковой информации используют микрофон, а для воспроизведения — акустические системы (звуковые колонки и наушники).



Рис. 2.7

- ? 1. Какие основные устройства содержит системный блок?
- 2. Какие устройства расположены на материнской плате?
- 3. Каково назначение процессора?
- 4. Для чего служат оперативная память и постоянное запоминающее устройство?
- 5. Какие устройства предназначены для длительного хранения информации?

## § 4. Операционная система

Обработка информации в компьютере и управление всеми его устройствами осуществляется с помощью программ. Программы, с которыми Вы знакомы, позволяют обрабатывать тексты и изображения.

Компьютер представляет собой совокупность программного и аппаратного обеспечения. При помощи программ и соответствующих устройств можно создавать музыку, распознавать и переводить тексты, управлять сложными машинами и технологическими процессами.

Набор всех установленных на компьютере программ называют его **программным обеспечением**. На каждом компьютере этот набор может быть разным. На домашних компьютерах устанавливают

текстовые и графические редакторы, игры, программы воспроизведения музыки и фильмов. На компьютере конструктора — сложные системы для выполнения чертежей и расчетов. Программное обеспечение определяется задачами, которые решает пользователь. Однако существуют программы, без которых компьютер работать не может.

Необходимым компонентом программного обеспечения компьютера является *операционная система* (ОС). **Операционная система** — это комплекс программ, позволяющих человеку вести диалог с компьютером, управлять всеми устройствами и программами. Операционная система автоматически загружается в оперативную память при включении компьютера. Только после этого компьютер готов к работе.

Главные задачи, которые решает операционная система:

- запуск и согласованное выполнение всех процессов и программ;
- управление хранением информации;
- управление работой всех устройств компьютера;
- взаимодействие пользователя с аппаратным и программным обеспечением компьютера.

Наиболее известными операционными системами являются MS DOS, MS Windows (XP, Vista), UNIX, Linux, Apple Mac OS.

Взаимодействие человека с компьютером выполняется с помощью команд. Способ, которым пользователь взаимодействует с компьютером, называется **пользовательским интерфейсом** или просто **интерфейсом**.

Интерфейс обеспечивают программные и аппаратные средства компьютера. Например, команды можно вводить с помощью клавиатуры, а результаты выводить на экран или принтер в виде текстов и чисел. Такой интерфейс называют **текстовым**. Речевой интерфейс, который не получил еще широкого распространения, позволяет управлять компьютером путем ввода голосовых команд с помощью микрофона.

Интерфейс большинства современных операционных систем является **графическим**. Рассмотрим графический интерфейс операционной системы Microsoft Windows XP.

Объекты графического интерфейса, которыми манипулирует пользователь, изображаются на мониторе в виде рисунков. Действия пользователя соответствуют определенным манипуляциям мышью. В мобильных компьютерах могут использоваться также устройства ввода в виде шарика (трекбол) или площадки, чувствительной к перемещению пальца (тачпад).

Основными элементами интерфейса являются Рабочий стол, Панель задач, меню кнопки Пуск, а также окна и значки, которые пользователь видит на экране монитора (рис. 2.8).

**Рабочий стол** — это изображение, которое появляется на экране монитора после включения компьютера и загрузки операционной системы. На Рабочем столе находятся Панель задач, кнопка Пуск, значки Мой компьютер, Корзина и другие элементы графического интерфейса Windows.

**Панель задач** представляет собой горизонтальную полосу в нижней части экрана. Слева на панели задач находится кнопка Пуск, открывающая

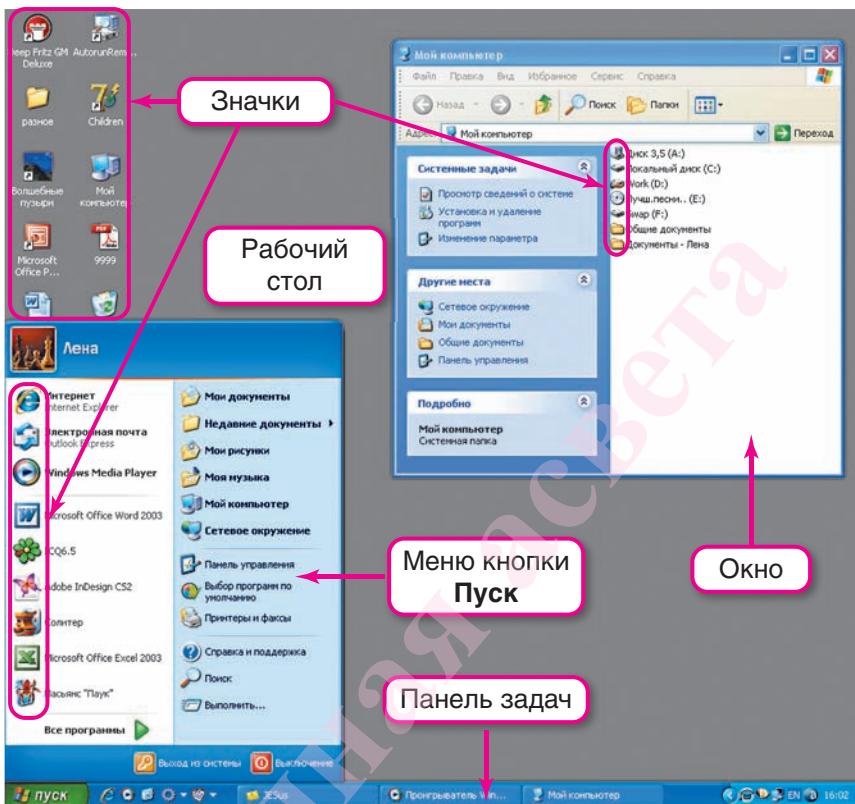


Рис. 2.8

главное меню. С его помощью осуществляется запуск программ, которые в Windows называют **приложениями**. Некоторые из приложений Вам известны, например графический редактор Paint или текстовый редактор Word. Главное меню служит также для настройки компьютера, получения справки, быстрого доступа к документам. Завершают работу компьютера также при помощи меню кнопки Пуск.

Основное назначение Панели задач — обеспечить быстрый запуск программ и доступ к открытым

тым окнам. Их значки и кнопки размещаются в средней части Панели задач. Справа находятся часы и значки, отображающие состояние некоторых программ и параметров компьютера, например переключение языка. При открытии окна программы или документа на Панели задач появляется соответствующая кнопка со значком и именем объекта.

**Окно** — один из самых важных объектов Windows. В окнах осуществляется ввод и вывод информации. Различают окна папок, рабочие окна приложений, диалоговые окна, окна справочной системы.

Одновременно могут быть открыты несколько окон. Windows является многозадачной операционной системой. Каждая задача (программа, документ) открывается в своем окне. Однако только одно окно может быть активным — то, в котором пользователь работает в данный момент времени. Его заголовок выделяется более ярким цветом. Другие окна являются неактивными. Чтобы переключиться на другое окно, нужно нажать соответствующую ему кнопку на Панели задач.

Окна, меню, кнопки, программы, папки, файлы и другие компоненты Windows называют одним термином — **объект**. Для обозначения объектов используются маленькие рисунки — **значки (иконки)**. Рядом со значениями может быть текст с информацией об объекте. Некоторые значки (их называют **ярлыками**) служат ссылками на объекты и используются для быстрого доступа к этим объектам.

Для настройки компьютера, установки и удаления программ используется **Панель управления**, переход к которой выполняется с помощью меню кнопки **Пуск**.

- ?
1. Для чего предназначена операционная система компьютера?
  2. Приведите примеры программ, составляющих программное обеспечение Вашего компьютера.
  3. Что называется пользовательским интерфейсом?
  4. В каком случае говорят, что окно активно?
  5. Для чего служат ярлыки?

### Упражнения

1. Назовите элементы интерфейса Windows, расположенные на Рабочем столе Вашего компьютера.
2. Перечислите названия изученных Вами ранее приложений, которые можно активизировать, выполнив последовательность команд **Пуск → Все программы**.

## § 5. Работа с окнами

Окно обычно содержит следующие элементы: строка заголовка, строка меню, Панель инструментов, адресная строка, рабочая область, строка состояния, полосы прокрутки.

Откроем окно **Мой компьютер** (рис. 2.9). Для этого выполним двойной щелчок мышью на ярлыке



, который находится на Рабочем столе (или воспользуемся меню кнопки **Пуск**).

В обычном состоянии окно занимает часть Рабочего стола. Окно можно развернуть на весь Рабочий стол или свернуть, оставив кнопку на Панели задач.

В правой части строки заголовка находятся три кнопки управления окном. Щелчок мышью на левой кнопке сворачивает окно, не прекращая ра-

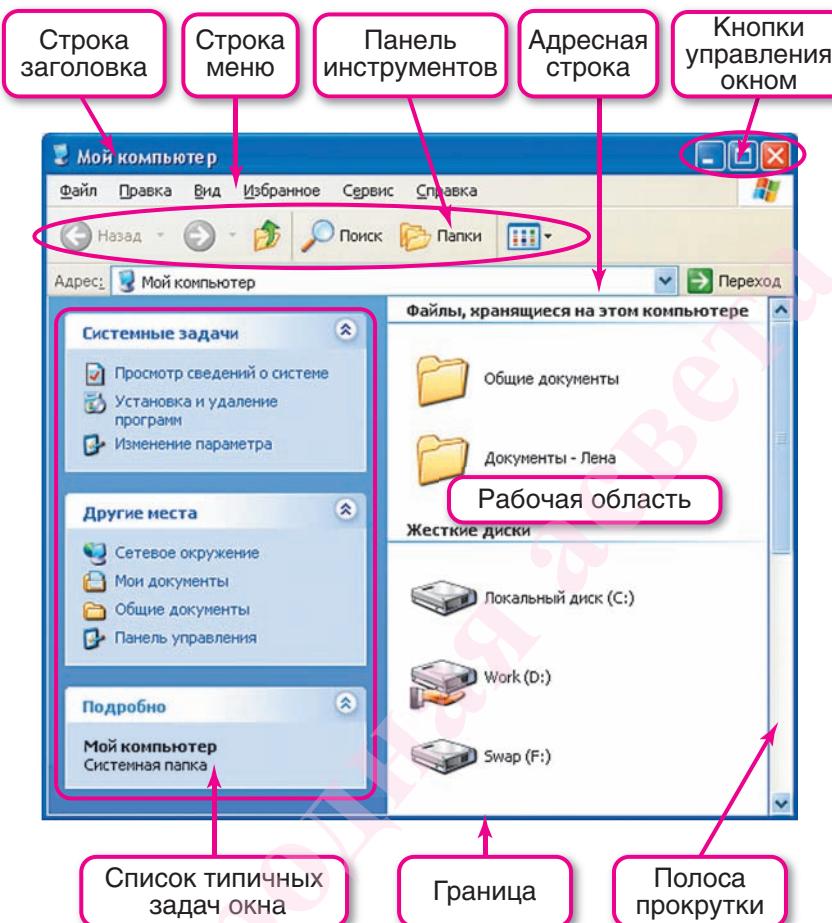


Рис. 2.9

боту приложения. Щелчок мышью на правой кнопке закрывает окно и завершает работу приложения. Вид средней кнопки и производимые действия зависят от состояния окна: нажатием кнопки окно из обычного состояния может быть развернуто на весь экран; нажатием кнопки развернутое окно может быть уменьшено до прежних размеров.

Положение и размеры окна в обычном состоянии можно изменять. Для перемещения окна необходимо установить указатель мыши на Строчке заголовка и, удерживая нажатой левую кнопку мыши, переместить окно в новое положение. Для изменения размеров окна достаточно перетащить его границы. Для этого подводят указатель мыши к выбранной границе до превращения курсора в двунаправленную стрелку и, удерживая нажатой левую кнопку мыши, передвигают границу.

Окна имеют выпадающие текстовые меню. Часто используемые команды могут выноситься на графические **Панели инструментов**. С использованием некоторых команд Вы уже познакомились при обработке графической и текстовой информации.

Способ отображения информации в окне выбирается с помощью меню **Вид** или **Панели инструментов**. Способ **Эскизы** удобен для просмотра изображений (рис. 2.10, *а*), способ **Таблицы** дает подробную информацию об объектах: размер файла, его тип, дату создания или изменения (рис. 2.10, *б*). Объекты можно упорядочить, например разместить их имена в алфавитном порядке, щелкнув мышью по заголовку списка.

Важную роль в интерфейсе Windows играют **диалоговые окна**. Они служат для организации диалога пользователя с операционной системой или приложениями. Диалоговые окна могут содержать текстовые поля, кнопки, списки, флажки и другие элементы, которые называют **элементами управления**.

Некоторые диалоговые окна Вам знакомы. Например, после подготовки с помощью редактора Paint

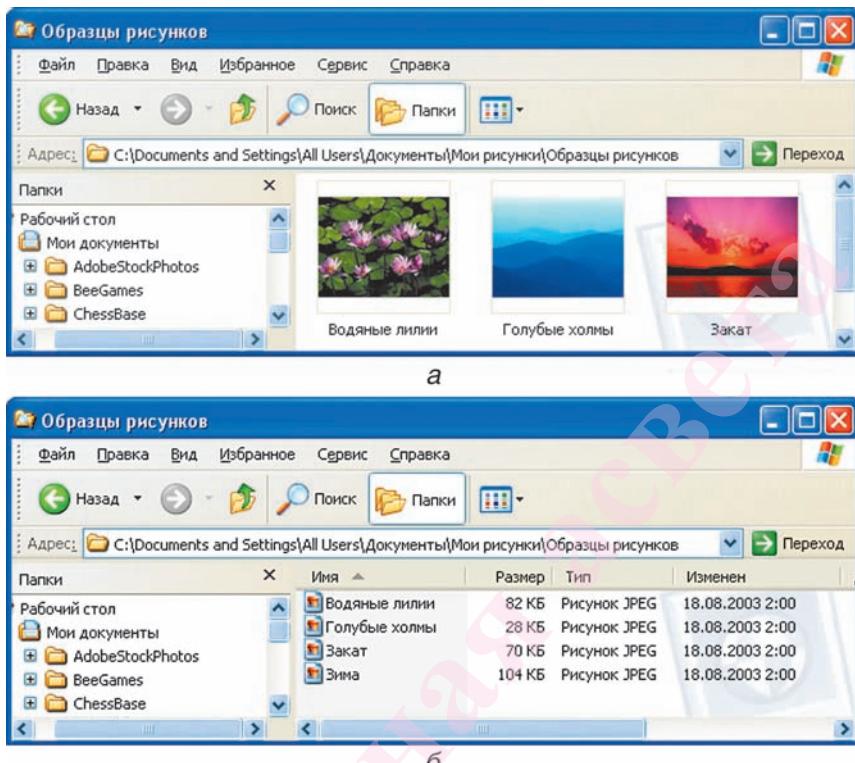


Рис. 2.10

рисунка его сохраняют. Для этого из меню **Файл** вызывают диалоговое окно **Сохранить как**, выбирают папку и вводят имя файла (рис. 2.11).

Помощь в работе с операционной системой и ответы на возникающие вопросы можно получить в справочной системе Windows, которая вызывается командами **Пуск → Справка и поддержка**.

Диалоговое окно **Центр справки и поддержки** (рис. 2.12) содержит текстовое поле **Найти**, в которое записывается краткая формулировка вопроса из одного или нескольких слов. Например, для получения справки об основных приемах работы с опе-

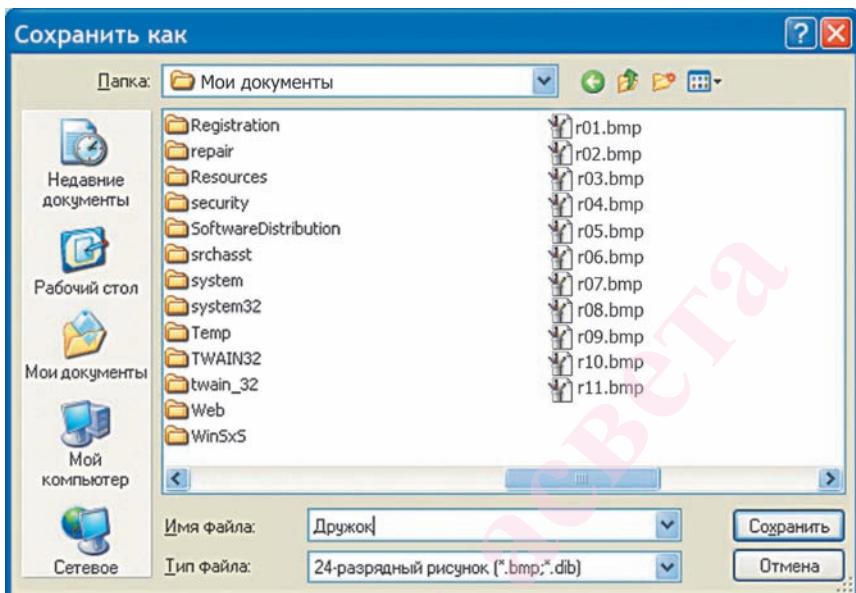


Рис. 2.11

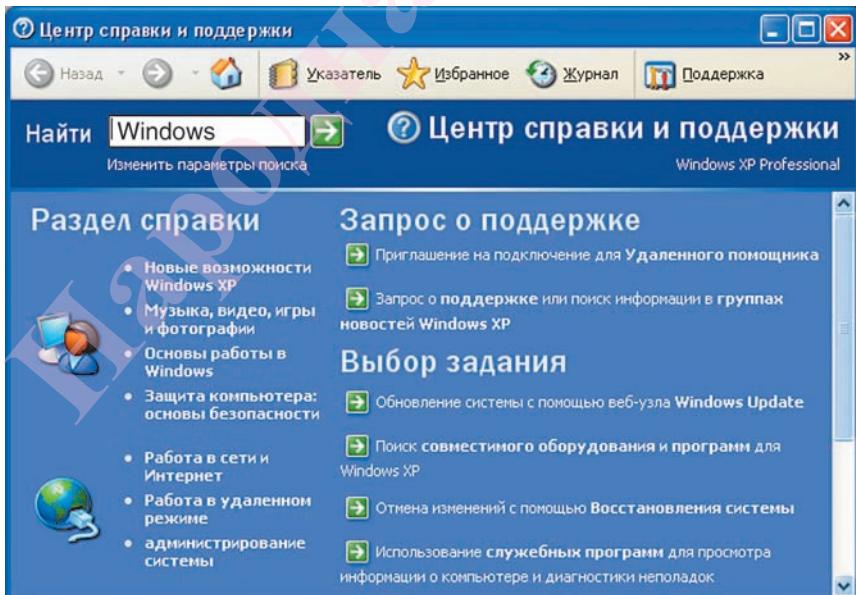


Рис. 2.12

рационной системой в текстовом поле введем слово **Windows** и нажмем кнопку ➔ (см. рис. 2.12).

Для управления компьютером и обработки информации в состав операционной системы Windows входят приложения, которые называют **стандартными**: текстовый и графический редакторы, программы для подключения к сети Интернет и др.

Например, стандартные приложения из меню **Связь** (рис. 2.13) используются для настройки сети. Приложения из меню **Служебные** позволяют архивировать данные, восстанавливать работу операционной системы или получать сведения о ней.

Разделы меню **Развлечения**, **Связь**, **Служебные**, **Специальные возможности** слева помечены значком , а справа — символом треугольник ►. Символ ► обозначает раскрывающийся список пунктов меню или программ (см. рис. 2.13).

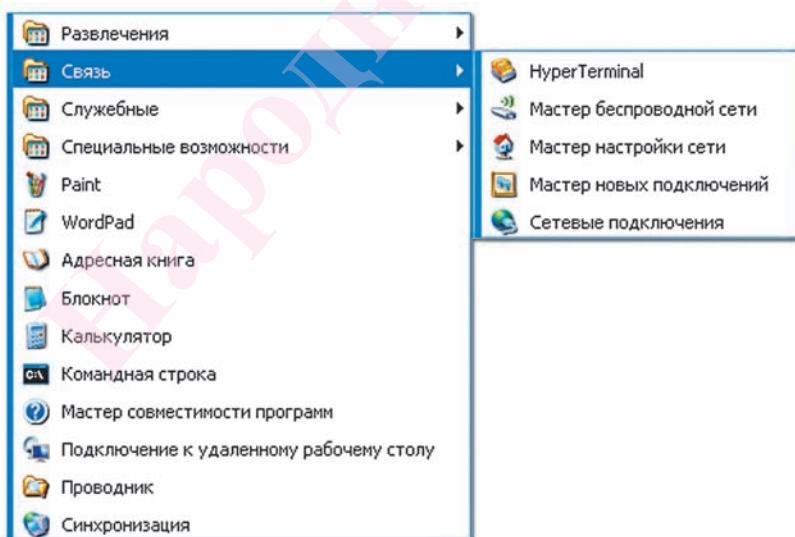


Рис. 2.13

Вызов стандартной программы управления звуковыми устройствами выполняется последовательностью команд: Пуск → Все программы → Стандартные → Развлечения → Громкость.

- ? 1. Можно ли изменить положение или размеры окна, которое развернуто на весь экран?
- 2. Как переместить окно на Рабочем столе?
- 3. Как изменить размеры окна?

### **Упражнения**

1. Разместите друг под другом окна **Мой компьютер** и **Мои документы**.
2. Разместите на Рабочем столе одновременно несколько копий окна **Мой компьютер** без наложения их друг на друга.
3. Получите в справочной системе Windows справку о Панели задач.
4. Назовите стандартные приложения, входящие в состав операционной системы, установленной на Вашем компьютере.

### **§ 6. Файловая система**

Одна из важнейших задач операционной системы — управление хранением информации.

Операционная система обеспечивает единообразное хранение различной информации (текстов, чертежей, фотографий, звуков, фильмов, программ) в виде файлов. Часть операционной системы, которая служит для работы с файлами, называется **файловой системой**.

Напомним основные свойства файлов. Ко всем файлам обращаются по именам. В Windows **имя файла** может состоять из последовательности симво-

лов общей длиной до 255 символов. В именах нельзя использовать символы: \ , / , : , \* , ? , < , > , | .

Имя файла может заканчиваться **расширением**, которое чаще всего состоит из трех символов. Расширение записывается в правой части имени файла после последней точки и указывает на тип файла. Например, **исполняемые файлы** (программы) имеют расширение .exe или .com.

Расширение (тип) файла	Содержание
.txt	Текст
.doc	Документ Word
.bmp, .gif, .jpg, .tif	Изображения
.ppt	Файлы презентаций PowerPoint
.wav, .mp3	Звуковые файлы
.avi, .mpeg	Видеофайлы
.pas	Программа на языке Pascal
.htm, .html	Веб-страницы
.exe, .com	Программы

Расширение файла помогает операционной системе определить, какая программа может использоваться для его обработки. Многие программы при сохранении файла сами приписывают расширение. Текстовые файлы, созданные в редакторе Блокнот, имеют расширение .txt, а созданные в редакторе Word — расширение .doc. Графическим файлам в цифровом фотоаппарате чаще всего присваивается расширение .jpg, а созданным в редакторе Paint — расширение .bmp.

Имена файлов следует выбирать так, чтобы отражалось их содержание и назначение. Например, **Портрет Адама Мицкевича.bmp**, **Реферат “История моего города”.doc** и др. Длинные имена файлов лучше отражают их содержание и назначение (сравните с **Портрет.bmp** и **Реферат.doc**).

Каждый файл характеризуется размером — объемом памяти, необходимым для его хранения. Например, файл **текст.txt**, содержащий страницу неформатированного текста из 2000 знаков, имеет размер около 2 Кбайт. Файл **ris.bmp** с цветным изображением размером 800 × 600 точек занимает около 1,4 Мбайт, а файл **1.wav** с несжатой звукозаписью длительностью 1 мин — около 10 Мбайт.

Файлы помещают в папки. Они, как и файлы, имеют имена. Папки обозначаются значком  . Внутри папки могут размещаться не только файлы, но и другие папки, а также иные объекты файловой системы. Например, внутри папки **Мой компьютер** размещены диски, папки, файлы, ярлыки.

Одни папки создает пользователь, другие, такие, как **Мой компьютер** или **Корзина**, создаются автоматически при установке операционной системы.

О файле или папке, которые помещены в некоторую папку, говорят, что они вложены в эту папку. Имя вложенного файла записывают после знака \ (обратный слэш). Запись **Мои рефераты\История города.doc** означает, что файл **История города**, имеющий расширение **.doc**, вложен в папку **Мои рефераты**. Запись **C:\Мои документы** начинается именем диска, на котором находится папка **Мои документы**.

**Путь к файлу (папке)** — это запись последовательности всех папок на пути от диска к нужному файлу или папке. Путь начинается латинской бук-

вой, обозначающей имя диска, с обязательным двоеточием после нее, содержит последовательность имен папок и заканчивается именем файла, например **C:\Мои документы\Реферат.doc**. Путь для выбранного файла (папки) отображается в адресной строке.

Пример 1. Записать путь к папке, которая будет открыта в результате последовательности команд: **Пуск → Мой компьютер → Локальный диск (С:) → → Program Files → Microsoft Office**.

Выполнив данную последовательность команд, мы откроем папку Microsoft Office. Путь к этой папке записывается так: **C:\Program Files\Microsoft Office**. Этот текст и показан в адресной строке открытого окна (рис. 2.14).

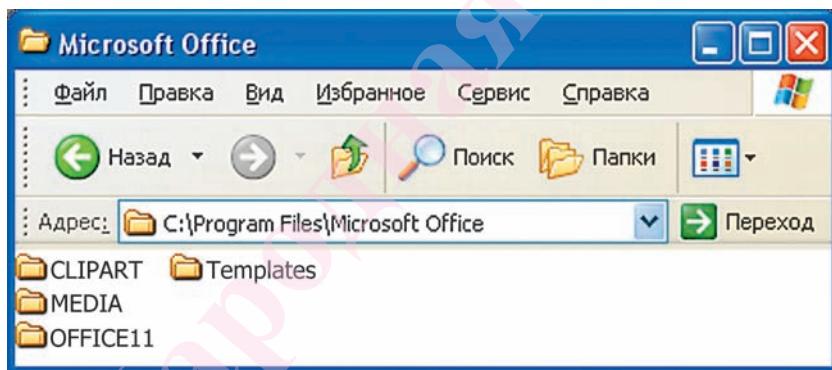


Рис. 2.14

Пример 2. Записать путь к папкам CLIPART, MEDIA, вложенным в папку Microsoft Office (см. рис. 2.14).

Путь к папке CLIPART:

**C:\Program Files\Microsoft Office\CLIPART.**

Путь к папке MEDIA:

**C:\Program Files\Microsoft Office\MEDIA.**

Заметим, что все папки, вложенные в папку **Microsoft Office**, имеют общую часть адреса: **C:\Program Files\Microsoft Office**.

Система размещения информации, при которой одни объекты файловой системы (файлы, папки, диски) вложены в другие, называется **иерархической**. По одному адресу (в одной папке) не могут находиться два объекта с одинаковыми именами.

- ? 1. Для чего служит файловая система?  
2. Какие расширения могут иметь файлы, содержащие изображения?  
3. Какую информацию содержат файлы с расширением .txt?  
4. Путь к файлу реферат.doc записан так:

**C:\work\7класс\реферат.doc.**

В какую папку вложен этот файл? На каком диске находится папка **7класс**?

### **Упражнения**

1. Получите список объектов папки **Мой компьютер** Вашего компьютера. Сколько дисков для хранения информации имеет Ваш компьютер?

2. Назовите папки, которые содержатся на диске **C:** Вашего компьютера.

3. Назовите файлы, расположенные в папке **Мои документы** Вашего компьютера.

4. Запишите путь к файлу по описанию: имя файла **Сочинение.doc**, этот файл содержится в папке **Мой город**, которая находится на диске **C:**.

5. Исправьте ошибки в записях имен файлов:

а) D:\тексты\\*новый\*.txt;

б) C:\works\ris>2.bmp;

в) C\ работа\фото\лес:дерево.jpg;

г) C:\ математика/ неравенство a < b.txt.

## §7. Операции с объектами файловой системы

### 7.1. Файловый менеджер Проводник. Поиск файлов и папок

Файловая система Windows позволяет выполнять следующие операции: поиск, создание, копирование, перемещение, переименование, удаление файлов и папок.

Программы для управления объектами файловой системы называют **файловыми менеджерами**. В операционную систему Windows входит файловый менеджер **Проводник**. Другими известными файловыми менеджерами являются Norton Commander, FAR Manager, Total Commander.

Для запуска **Проводника** нажатием правой кнопки мыши вызовем контекстное меню кнопки **Пуск** и выберем пункт **Проводник**. Второй способ запуска — выполнение последовательности команд: **Пуск → Все программы → Стандартные → Проводник**.

Окно **Проводника** имеет две панели. На левой панели показываются папки и диски, а на правой панели — объекты, которые содержатся в выбранной папке (рис. 2.15). Некоторые папки, например **Моя музыка**, **Локальный диск (С:)**, помечены знаком +. Это означает, что они содержат вложенные папки.

С помощью **Проводника** удобно обозревать файловую структуру (дерево папок) и проводить поиск объектов. Так, на диске **D:** находится папка **Работа**, в ней — папка **Склад** с папками: **Документы**, **Портреты**, **Рисунки**, **Тексты** (рис. 2.16).

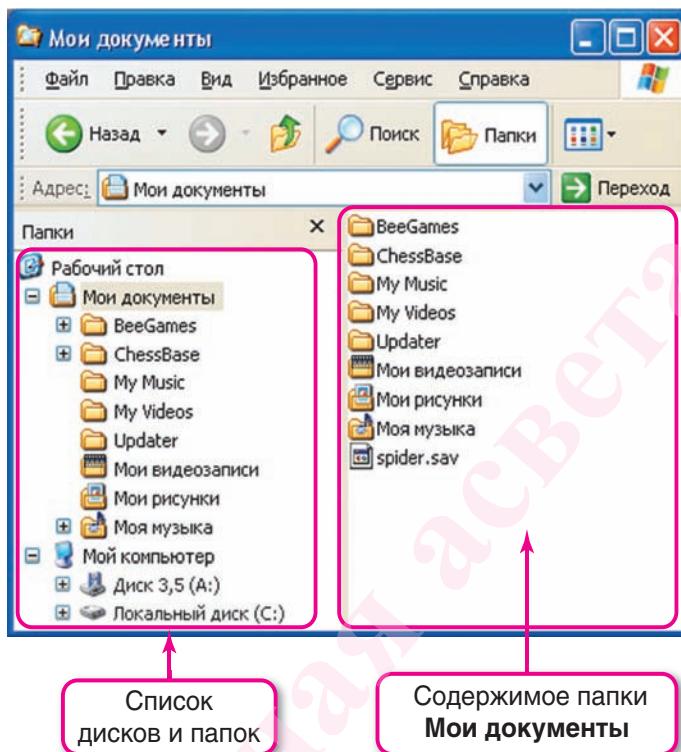


Рис. 2.15

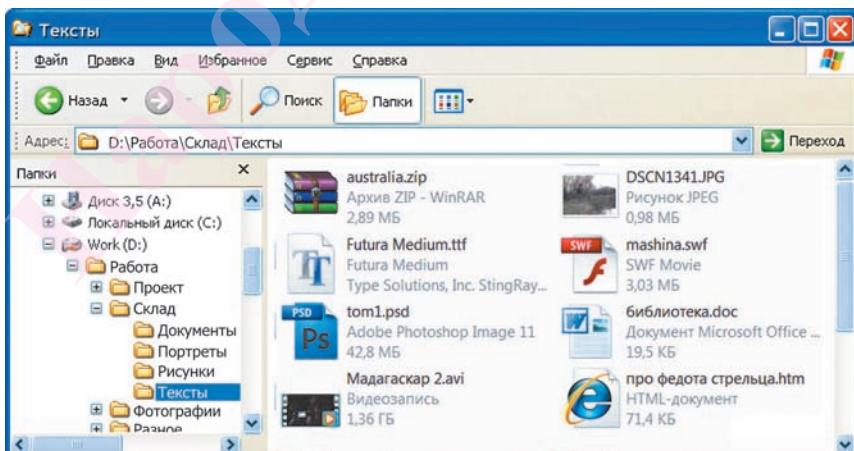


Рис. 2.16

Способ отображения информации в окне выбирается с помощью меню **Вид** или Панели инструментов. Например, при выборе способа **Плитка** вид соответствующих значков указывает на тип файлов.

Система Windows располагает специальными средствами поиска, благодаря которым файл или папку можно найти по имени, даже если неизвестен путь.

**Пример 1.** На диске С: найти папку **Мои документы**.

Щелкнем мышью на кнопке **Поиск** Панели инструментов **Проводника**. В появившемся диалоговом окне **Поиск файлов и папок** введем текст запроса **Мои документы** и укажем диск для поиска С: (рис. 2.17).

В окне **Результаты поиска** будет выведена информация о двух найденных папках с именем **Мои документы**, которые вложены в разные папки (рис. 2.18).

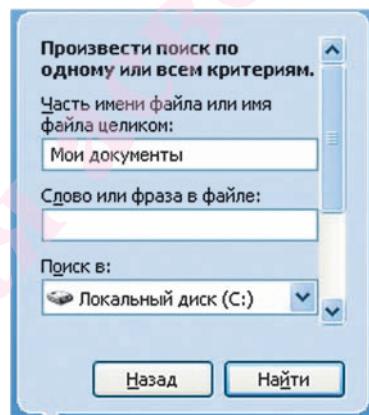


Рис. 2.17

Имя	Папка	Размер	Тип	Изменен
Мои документы	C:\Documents and Settings\Лена		Папка с файлами	05.06.2009 9:21
Мои документы	C:\WINDOWS\system32\config...		Папка с файлами	02.12.2008 23:43

Рис. 2.18

Выбирая необходимые объекты и настраивая параметры, можно осуществить поиск группы файлов определенного типа с заданными свойствами.

В качестве параметров поиска можно задавать дату создания (изменения), размер и тип файлов.

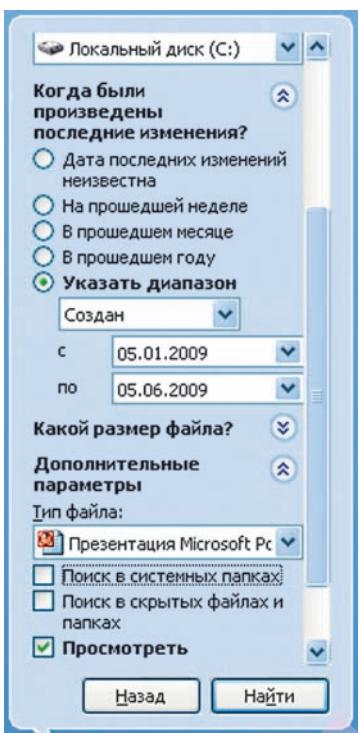


Рис. 2.19

текст. Например, с ее помощью можно найти все файлы, содержащие слово «информатика».

## 7.2. Создание и переименование папок и файлов

Рассмотрим на примерах, как можно создавать и переименовывать файлы и папки.

**Пример 3.** В папке **Мои документы** создать папки с именами **Моя работа** и **Примеры**.

На левой панели **Проводника** выберем папку **Мои документы**. Выполним команды: **Файл → Создать → Папку** (рис. 2.20). Вместо предлагаемого имени **Новая папка** введем новое имя **Моя работа**. Нажмем клавишу **Enter** или левую кнопку мыши.

**Пример 2.** На диске С: найти все файлы презентаций PowerPoint, созданные за последние 6 месяцев.

Установим параметры поиска. Укажем дату создания файла: зададим диапазон с **05.01.2009** по **05.06.2009** — и тип файла **Презентация Microsoft PowerPoint** (рис. 2.19).

В окне **Результаты поиска** будет выведена информация обо всех найденных презентациях, удовлетворяющих требуемым параметрам.

Еще одна интересная возможность — поиск файла, содержащего заданный текст.

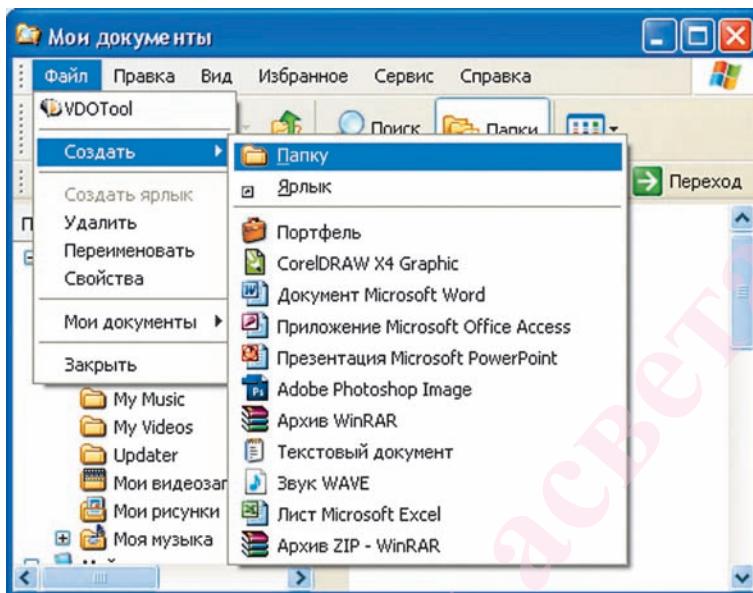


Рис. 2.20

Аналогичным способом создадим папку **Примеры**.

**Пример 4.** В папке **Моя работа** создать текстовый файл с именем **Данные**.

На левой панели **Проводника** выберем папку **Моя работа**. Выполним команды: **Файл** → **Создать** → **Текстовый документ** (рис. 2.21). Вместо предлагаемого имени **Текстовый документ.txt** введем имя **Данные**, не меняя расширения **.txt**. Нажмем клавишу **Enter**.

**Пример 5.** Изменить имя файла **Данные** на **Пример текста**.

На правой панели **Проводника** выберем файл с именем **Данные**. Выполним команды: **Файл** → **Переименовать** (рис. 2.22). Вместо старого имени **Данные.txt** напишем **Пример текста.txt**. Нажмем клавишу **Enter**.

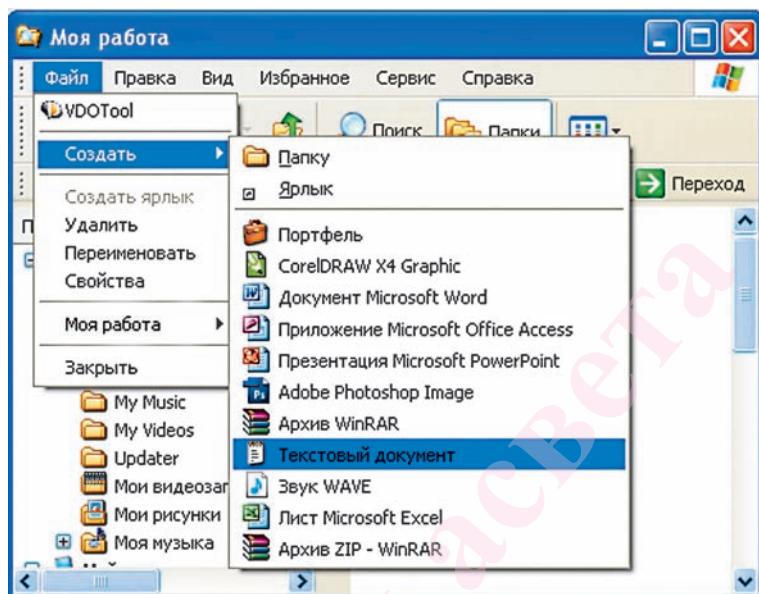


Рис. 2.21

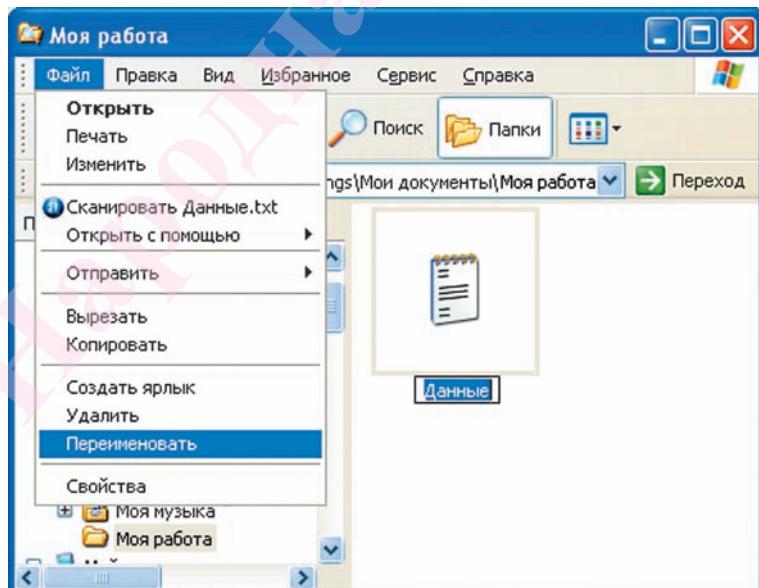


Рис. 2.22

Аналогичным способом изменяют имена папок. Следует помнить, что переименовать используемый другим приложением Windows файл нельзя.

### 7.3. Копирование, перемещение и удаление файлов и папок

**Копирование файла (папки)** — это создание копии файла (папки) в другой папке. Источником является папка с исходным файлом, приемником — другая папка для размещения копии файла. После копирования получают два одинаковых файла, расположенных в разных папках.

Пример 6. Скопировать файл **Пример текста.txt** из папки **Моя работа** в папку **Примеры**.

Выделим файл **Пример текста.txt** в папке **Моя работа**. Выполним команды **Правка → Копировать** (рис. 2.23). На левой панели **Проводника** выберем папку **Примеры**. Выполним команды **Правка → Вста-**

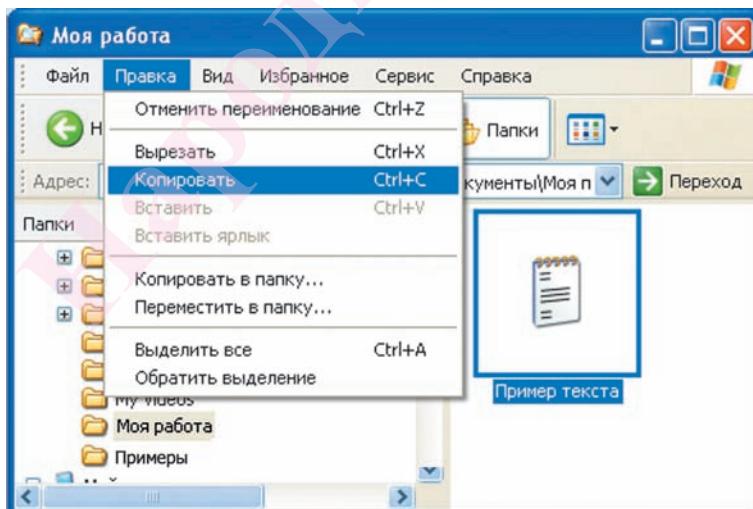


Рис. 2.23

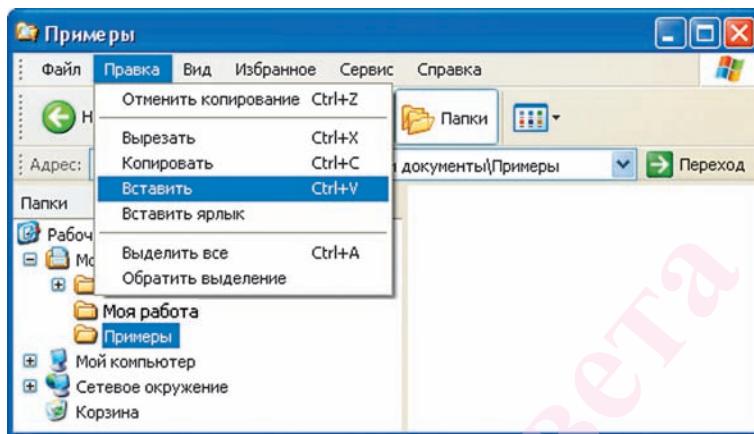


Рис. 2.24

вить (рис. 2.24). В результате этих действий копия файла **Пример текста.txt** будет размещена в папке **Примеры**, а оригинал останется в папке **Моя работа**.

**Перемещение файла** — это создание копии файла в папке-приемнике без сохранения его в папке-источнике. Перемещение объекта выполняется аналогично копированию. В папке-источнике выбирают объект для перемещения и выполняют команды **Правка** → **Вырезать**. Выбирают папку-приемник и выполняют команды **Правка** → **Вставить**.

Копировать или перемещать можно не все объекты. Используя файловую систему, невозможно создать копию или переместить папки, соответствующие дискам, или папку **Мой компьютер**.

Ненужные файлы и папки могут быть удалены. Для этого их выделяют, а затем нажимают клавишу **Del** на клавиатуре или выполняют команду **Файл** → → **Удалить**.

**Пример 7.** Удалить файл **Пример текста.txt**.

Выделим файл **Пример текста.txt** в папке **Примеры**. Нажмем клавишу **Del**.

После удаления объекты обычно помещаются в **Корзину**. **Корзина** предназначена для временного хранения удаленных объектов. Объекты из **Корзины** могут быть восстановлены.

**Пример 8.** Восстановить удаленный файл **Пример текста.txt**.

Откроем Корзину. В окне Корзина выделим файл **Пример текста.txt**. Выполним команды **Файл → Восстановить** (рис. 2.25).

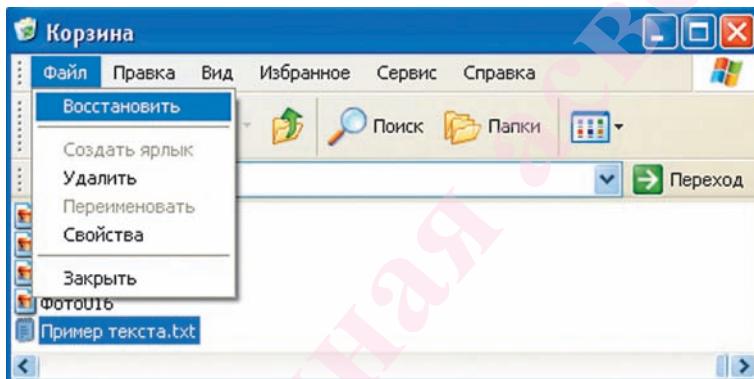


Рис. 2.25

Объекты из **Корзины** также можно удалить. Удаленные из **Корзины** объекты восстановить с помощью операционной системы невозможно.

Для выполнения операций сразу с несколькими объектами, все их нужно выделить. Если объекты расположены рядом, то их выделяют так: щелкают мышью на первом объекте и, удерживая клавишу **Shift**, на последнем. Если выделяемые объекты не расположены рядом, то их выделяют, удерживая клавишу **Ctrl**. Для выделения всех объектов внутри папки используют команды **Правка → Выделить все**.

В примерах 1—6 мы использовали команды меню **Проводника**.

Операции с файлами и папками можно выполнять и другими способами, например использовать контекстное меню, вызываемое правой кнопкой мыши.

Перемещение объекта можно осуществить перетаскиванием его значка из папки-источника в папку-приемник с нажатой левой кнопкой мыши. Для этого должны быть открыты два окна с этими папками. При перетаскивании значка с нажатой правой кнопкой мыши можно сделать выбор операции: копирование или перемещение. Для удаления объекта достаточно перетащить его значок в **Корзину**.

Ускоряет работу использование комбинаций клавиш, например Ctrl + C (копировать), Ctrl + X (вырезать), Ctrl + V (вставить), Ctrl + A (выделить все).

**Пример 9.** На Рабочем столе создать папку **Работа** и текстовый файл **Проект.txt**. Скопировать в папку **Мои документы** папку **Работа** и файл **Проект.txt**. Удалить их с Рабочего стола.

Щелкнем правой кнопкой мыши на Рабочем столе. В контекстном меню выберем: **Создать** → **Папка** и введем имя **Работа**.

Еще раз щелкнем правой кнопкой мыши на Рабочем столе. В контекстном меню выберем: **Создать** → **Текстовый документ** и введем имя **Проект**.

Откроем окно с папкой **Мои документы** (можно использовать **Проводник** или папку **Мой компьютер**). Перетащим в нее папку **Работа** и файл **Проект.txt** с нажатой правой кнопкой мыши, выбирая операцию **Копировать**.

Для удаления объектов с Рабочего стола перетащим их в Корзину.

#### 7.4. Создание ярлыков

Для быстрого доступа к программам, файлам и другим объектам файловой системы используются ярлыки — небольшие файлы, содержащие рисунок-пиктограмму и путь к заданному объекту. Рисунки ярлыков помечаются знаком  в левом нижнем углу.

Ярлыки являются объектами файловой системы. Их можно создавать и удалять, копировать и перемещать.

Для создания ярлыка необходимо: открыть папку для его размещения и выполнить последовательность команд: **Файл** → **Создать** → **Ярлык** → **Обзор** → → **Показать размещение объекта** → **Далее** → **Имя ярлыка** → **Готово**.

Для создания ярлыка на Рабочем столе используется контекстное меню.

**Пример 10.** На Рабочем столе создать ярлык для доступа к папке **C:\Мои документы**.

Нажмем правую кнопку мыши в области Рабочего стола. В контекстном меню выберем **Создать** → **Ярлык** (рис. 2.26). Укажем место размещения объекта

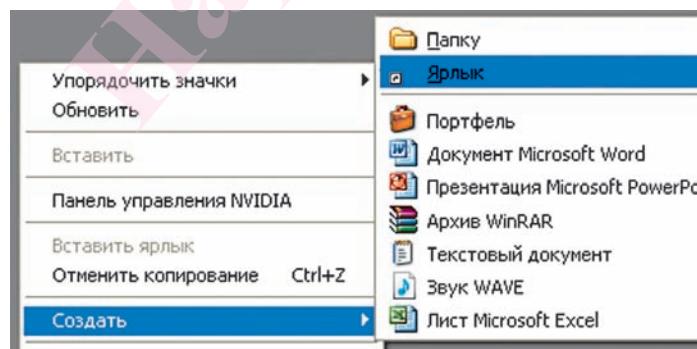


Рис. 2.26

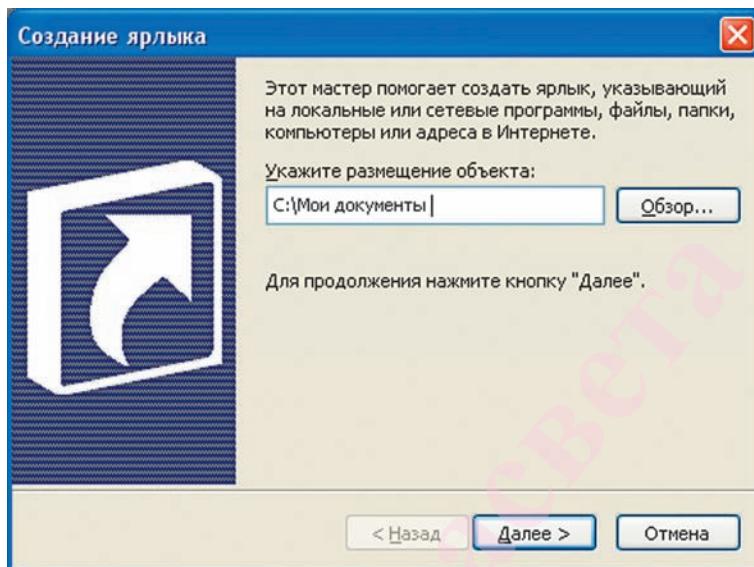


Рис. 2.27

с помощью кнопки **Обзор** (рис. 2.27). Далее введем имя ярлыка, например **Папка Мои документы**.

Заметим, что удаление ярлыка не приводит к удалению объекта, на который он указывает.

- ?
1. Для чего служит файловый менеджер?
  2. Как найти объект, зная его имя?
  3. Какие операции можно производить с файлами и папками?
  4. Для чего предназначена Корзина?
  5. Как создать ярлык на Рабочем столе?

### **Упражнения**

1. Покажите папки, которые содержатся на диске D: Вашего компьютера.
2. Покажите папки и файлы, расположенные в папке **Мои документы** Вашего компьютера.

3. На диске С: найдите все графические файлы с расширением .bmp, созданные за последние 12 месяцев.

4. Найдите все текстовые документы с расширением .doc, размер которых не превышает 100 Кбайт.

5. В папке **Мои документы** создайте папку с именем, обозначающим Ваш класс. В этой папке создайте папки для размещения работ пяти одноклассников. Имена папок одноклассников задайте в соответствии с их именами и фамилиями. Возможный результат выполнения упражнения изображен на рисунке 2.28, а.

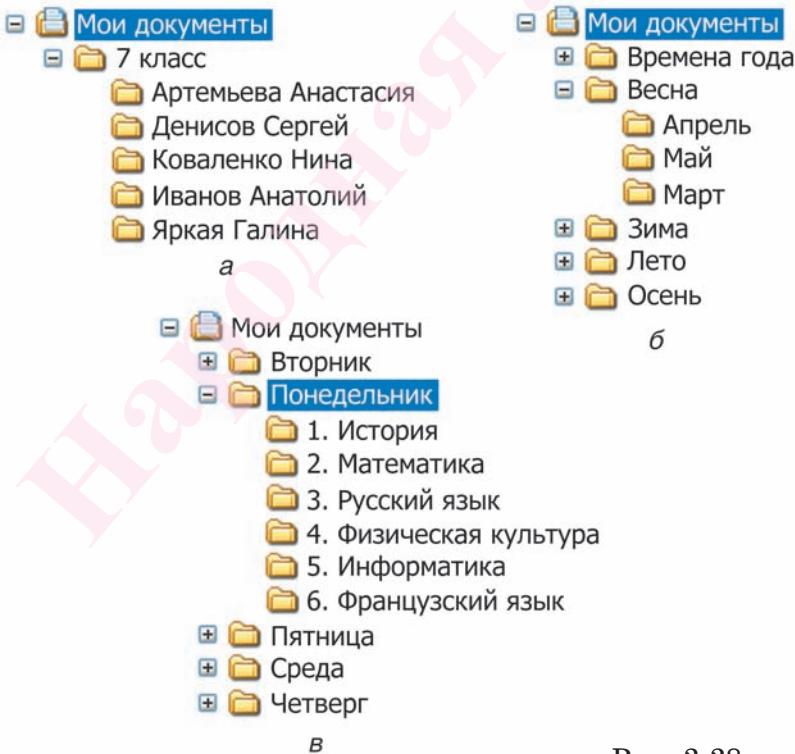


Рис. 2.28

6. В папке **Мои документы** создайте папку с именем **Времена года**. В этой папке разместите папки **Весна, Зима, Лето, Осень**. Внутри каждой из папок с названиями времен года разместите папки с именами месяцев для каждого времени года. Возможный результат выполнения упражнения представлен на рисунке 2.28, б.

7. В папке **Мои документы** создайте папку **Расписание уроков**. В этой папке создайте пять папок с именами **Понедельник, Вторник, Среда, Четверг, Пятница**. Внутри каждой из этих папок разместите папки с именами — названиями уроков, которые проводятся в этот день. Возможный результат выполнения упражнения изображен на рисунке 2.28, в.

## §8. Локальная компьютерная сеть

Для оперативного обмена информацией и совместного использования общих ресурсов компьютеры объединяют в сеть. **Ресурсами** сети могут быть устройства (принтеры, сканеры, видеокамеры, проекторы) и информационные объекты (программы, файлы, папки, базы данных).

**Локальная компьютерная сеть** представляет собой объединение компьютеров, расположенных на небольших расстояниях друг от друга.

Компьютеры локальной сети могут быть расположены в пределах одной организации, предприятия, учебного заведения. Например, компьютеры, размещенные в одном компьютерном классе, целесообразно объединять в локальную сеть.

**Рабочей группой** называют группу компьютеров, сообща использующих общедоступные для

этой группы ресурсы, например общие папки, принтеры.

Для работы компьютеров в сети необходимо соответствующее аппаратное и программное обеспечение.

Аппаратное обеспечение локальной сети включает сетевые адAPTERы компьютеров и соединительные кабели. Современным компьютерам отдельный сетевой адAPTER не требуется, он размещен на материнской плате. Компьютеры в сети соединяются с помощью кабеля. Однако все чаще используются беспроводные локальные сети.

Программную поддержку работы компьютеров в локальной сети выполняет операционная система.

Рассмотрим на примере операционной системы Windows, как осуществляется доступ, просмотр и использование ресурсов локальной сети.

При загрузке системы Windows для входа в компьютерную сеть в специальном диалоговом окне необходимо указать имя пользователя и его пароль.

Просмотр доступных ресурсов сети (сетевых ресурсов) осуществляется в папке **Сетевое окружение**. В окне этой папки отображаются общие ресурсы сети, к которой подключен компьютер (компьютеры, папки, файлы, принтеры).

Откроем папку **Сетевое окружение** с помощью **Проводника** или двойного щелчка левой кнопкой

мыши на значке **Сетевое окружение**



Папка **Сетевое окружение** содержит имена рабочих групп, например **Cabinet21**, **Cabinet22**, а так-

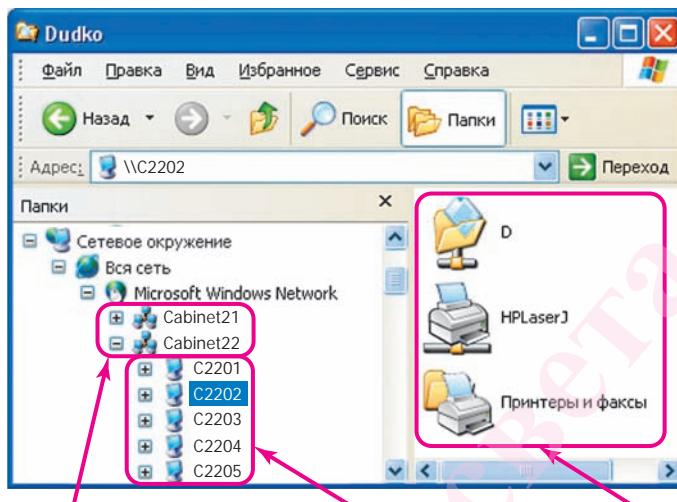


Рис. 2.29

же имена доступных в сети компьютеров, например **C2201**, **C2202** (рис. 2.29).

При щелчке кнопкой мыши на значке компьютера, например **C2202**, пользователь может увидеть, какие ресурсы сети, находящиеся на этом компьютере, ему доступны. В нашем примере (см. рис. 2.29) — это диск **D**; папка **Принтеры и факсы** и принтер **HPLaserJ**.

Пользователь, на компьютере которого находится ресурс (файл, диск, папка или устройство), является его владельцем и имеет полный доступ к этому ресурсу. Владелец ресурса может разрешить другим пользователям сети доступ к своему диску, папке, файлу.

**Пример 1.** Обеспечить доступ пользователей локальной сети к папке **test**.

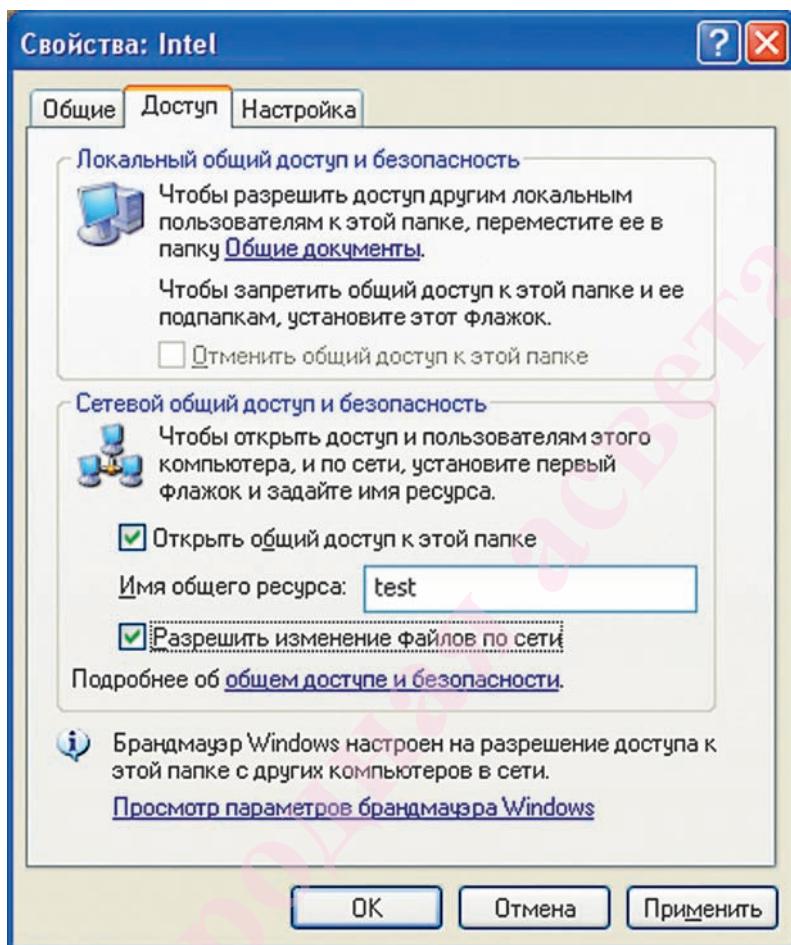


Рис. 2.30

Щелчком правой кнопкой мыши на объекте (папке **work**) в Проводнике вызовем контекстное меню и выберем пункт **Свойства**.

В окне **Свойства** (рис. 2.30) установим флажок **Открыть общий доступ к этой папке**. Введем сетевое имя ресурса (например, **test**). Теперь пользователи сети смогут читать содержимое этой папки, но не смогут его копировать.

В этом же окне можно установить флажок **Разрешить изменение файлов по сети**. Это обеспечит пользователям сети полный доступ к ресурсу.

Пользователь, для которого разрешен полный доступ к диску, папке, файлу в сети, может открывать, копировать, перемещать и удалять эти объекты. Это позволяет выполнять коллективную работу с общими ресурсами. Например, пользователи, имеющие полный доступ к файлу на одном из компьютеров сети, могут скопировать его на свои компьютеры с помощью операций **Копировать** и **Вставить** в Проводнике.

Владелец ресурсов в сети может запретить их общее использование. Для этого он должен установить флажок **Отменить общий доступ к этой папке** на вкладке **Доступ** окна Свойства.

Если к одному из компьютеров локальной сети подключен принтер, можно предоставить доступ к нему другим компьютерам сети.

**Пример 2.** Обеспечить доступ пользователей локальной сети к принтеру.

Откроем папку **Принтеры** в окне Проводника.

Щелчком правой кнопкой мыши на изображении нужного принтера вызовем контекстное меню и выберем пункт **Общий доступ**.

Установим переключатель **Общий доступ к данному принтеру**.

Документы, направленные пользователями на общесетевой принтер, выстраиваются в очередь на вывод в порядке их поступления (рис. 2.31).

Единых правил поведения пользователей в локальной сети не существует. Отметим лишь некоторые общие требования:

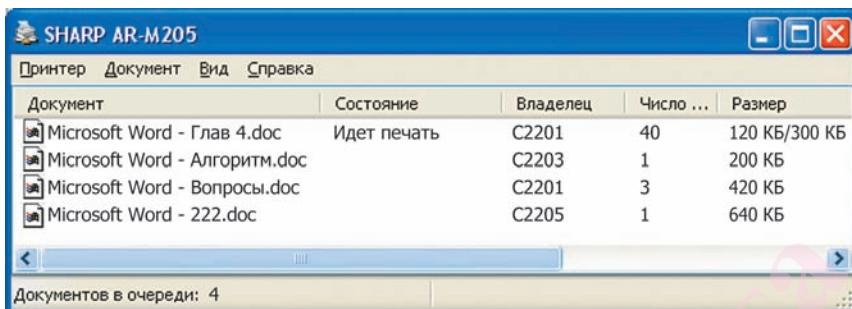


Рис. 2.31

- не передавайте другим пользователям ваше имя и пароль для входа в сеть;
  - по возможности сохраняйте информацию на диске Вашего компьютера, а не на дисках общего пользования.
- ? 1. Что собой представляет локальная компьютерная сеть?  
2. Что понимают под ресурсами локальной сети?  
3. С помощью какой папки можно просмотреть ресурсы сети?

# Глава 3

## ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

### §9. Среда программирования Pascal ABC

#### 9.1. Что такое язык программирования?

Вы уже знаете, что компьютер обладает поистине неограниченными возможностями: с ним можно поиграть в какую-нибудь игру, с его помощью можно написать сочинение, его можно превратить в музыкальный инструмент, в альбом для рисования и т. д.

Но все это возможно при одном условии: компьютер выполнит задание, если ему будет задана определенная последовательность команд. Составление последовательности команд для решения конкретных задач на специальном языке программирования называется разработкой программ или **программированием**.

Независимо от того, станет ли программирование Вашим увлечением либо профессиональным занятием, научившись составлять программы, Вы получите удовольствие от возможности самостоятельно управлять компьютером.

Язык программирования представляет собой набор слов, специальных знаков и команд, понятных компьютеру. Приступая к разработке программ, необходимо познакомиться с этими словами, а также с синтаксическими правилами, по которым они складываются в предложения программы.

Предложения, из которых состоит программа, набираются на клавиатуре. После этого компьютер переводит их в машинный код (на машинный язык), который состоит из нулей и единиц. Для выполнения такого перевода компьютеру нужен так называемый языковой процессор (компилятор), который встроен

в систему программирования. Система программирования Pascal ABC, с которой Вы продолжите знакомство, позволяет программировать на языке Pascal.

## 9.2. Структура программы

Программа на языке программирования Pascal оформляется стандартным образом и имеет следующий вид:

```
Program <имя программы>;  
<раздел описаний, в котором описываются  
данные>;  
Begin  
  <раздел команд (тело программы)>;  
End.
```

Рассмотрим на примере правила оформления программы.

Пример 1. Написать программу, которая выведет на экран монитора текст.

Запишем программу на языке Pascal.

```
Program text1; {Заголовок программы}  
Begin {Начало программы}  
  Write('Я всемогущий компьютер! Жду ваших  
  команд!');  
End. {Конец программы}
```

Здесь **Program text1;** — заголовок программы, где **text1** — имя программы (его пользователь задает самостоятельно). Желательно, чтобы имя программы соответствовало ее содержанию. Оно может содержать до 255 латинских букв, цифр, знаков подчеркивания (\_) и должно начинаться с буквы или знака подчеркивания. В качестве имен программ либо данных (величин) нельзя использовать имена

зарезервированных в языке программирования слов (например, Begin, End и др.). Эти слова даны в таблице 1 Приложения.

После каждого предложения в программе на языке Pascal нужно ставить точку с запятой.

После заголовка программы следует тело программы, которое начинается словом Begin. После него не ставят точку с запятой.

Команда Write выводит на экран монитора текст, заключенный в апострофы в скобках.

Словом End обозначается конец программы. После него обязательно ставят точку.

! При вводе текста программы с клавиатуры не имеет значения, какие используются буквы: строчные либо прописные.

В фигурных скобках можно писать комментарии (пояснительный текст, который облегчает понимание программы и не влияет на ее выполнение). Использование в программе комментариев необязательно.

### 9.3. Запуск среды Pascal ABC

Для вызова среды Pascal ABC необходимо запустить на выполнение файл **PascalABC.exe** либо щелкнуть мышью на значке Pascal ABC, если он есть на Рабочем столе.

При успешном запуске на экране появится главное окно программы (рис. 3.1). Оно подобно окнам известных Вам приложений, например текстового редактора Word либо графического редактора Paint.

При вводе и редактировании текста программы используются такие же приемы, как и при работе в текстовом редакторе.

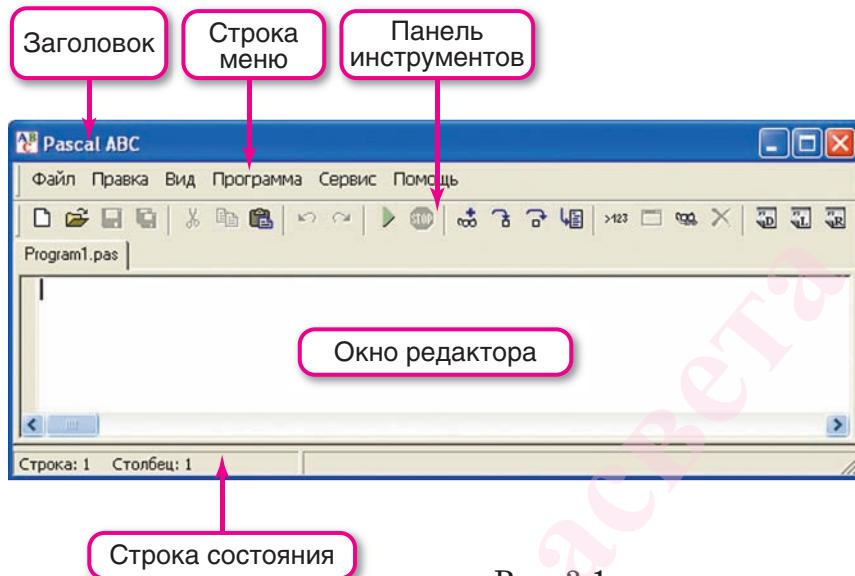


Рис. 3.1

С помощью мыши (либо клавиши F10 и клавиш управления курсором) можно открыть любой пункт меню.

#### 9.4. Работа в окне редактора

Для того чтобы набрать текст программы, т. е. создать новый файл, нужно выполнить следующие действия:

1. Выбрать меню **Файл**.
2. Выбрать команду **Новый**.

На экране откроется пустое окно редактора, озаглавленное именем **Program1.pas**. Это имя дается средой по умолчанию файлу, в котором будет храниться будущая программа. Его можно изменить по своему усмотрению при сохранении программы.

3. Набрать текст программы в окне редактора.

В этом окне можно не только вводить текст программы, но и редактировать его (удалять и встав-

лять символы и строки, «разрезать» и «склеивать» строки, искать и заменять текст).

Для перемещения курсора в окне редактора используются клавиши Home, End, PageUp, PageDown, для удаления текущей строки — Ctrl + Y, для удаления символа слева от курсора — Backspace, справа — Del, для задания отступов в строке — Tab.

При работе в текстовом редакторе системы программирования PascalABC можно использовать и другие комбинации клавиш, назначение которых приведено в таблицах 2, 3 Приложения.

С помощью команды **Помощь** → **Содержание** можно вызвать справочную систему Pascal ABC.

### 9.5. Сохранение и загрузка программ

Для сохранения введенной в окне редактора программы необходимо выполнить команду **Файл** → **Сохранить как...** → указать имя файла для сохранения → **Сохранить**.

Для загрузки программы из файла на диске нужно выполнить команду **Файл** → **Открыть** → выбрать нужный файл → **Открыть**.

### 9.6. Выполнение программы

Для выполнения набранной либо открытой программы необходимо выполнить команду **Программа** → **Выполнить** (либо нажать клавишу F9).

Если в программе отсутствуют ошибки, Вы увидите результат ее работы на экране монитора. В частности, в результате выполнения программы из Примера 1 на экране монитора в окне вывода появится следующий текст:

Я всемогущий компьютер! Жду ваших команд!

На рисунке 3.2 показано рабочее окно программы. В окне редактора находится исходный текст программы, а в окне вывода — результат ее выполнения.

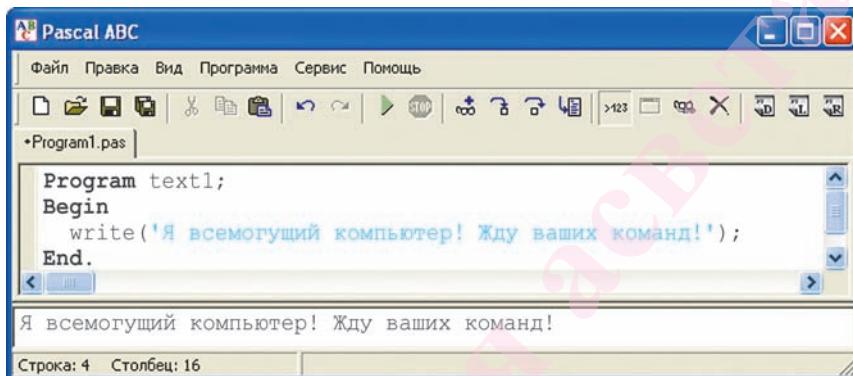


Рис. 3.2

Если в программе обнаружена ошибка, в окне редактора в области ошибки устанавливается курсор, а в окне вывода выдается сообщение об ошибке (рис. 3.3). Ошибку в тексте программы необходимо

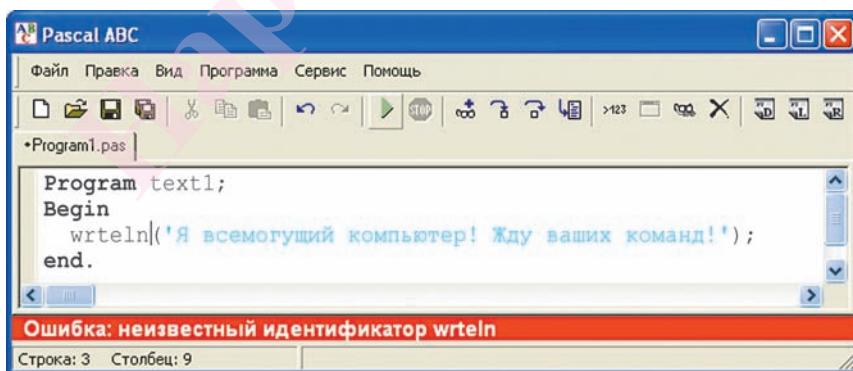


Рис. 3.3

найти и исправить. После этого можно снова попытаться выполнить программу.

В поиске ошибок Вам поможет среда Pascal ABC.

Различают *синтаксические и логические ошибки*.

**Синтаксические** ошибки возникают при нарушении правил языка программирования (неправильная запись команды, пропуск апострофов либо точек с запятой, отсутствие зарезервированных слов *Begin* (в начале программы), *End* (в конце программы), точки в конце программы, неверный формат команды и т. п.), их обнаруживает компилятор. В таблице 4 Приложения приведены сообщения о часто допускаемых начинающими программистами синтаксических ошибках.

**Логические** ошибки могут не прерывать выполнение программы, но приводят к неверным результатам. К числу логических ошибок относится неправильный алгоритм решения задачи либо использование неподходящей команды. Такие ошибки должен обнаруживать сам программист, например, с помощью пошагового выполнения программы (команда *Отладка*).

- ?
- 1. Какими зарезервированными словами начинается и заканчивается тело любой программы?
- 2. Какова структура программы?
- 3. Как загрузить среду Pascal ABC?
- 4. Как в среде Pascal ABC запустить программу на выполнение?

### *Упражнения*

1. Напишите программу, которая выводит на экран дату Дня Республики Беларусь.
2. Напишите программу, которая выводит на экран фразу «Моя Родина — Беларусь».

## § 10. Основные понятия языка программирования

### 10.1. Понятие переменной

Информацию, обрабатываемую компьютером или полученную в качестве промежуточных либо итоговых результатов, называют **данными**.

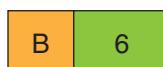
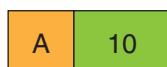
Данные в программировании принято называть **величинами**. Величины, значения которых изменяются в процессе выполнения программы, называют **переменными**, а те величины, которые не изменяют своего значения, — **константами**.

Значения величин, которые обрабатывает программа, должны находиться в оперативной памяти компьютера, состоящей из ячеек (байтов). Каждая ячейка имеет свой адрес (порядковый номер). Но с адресами в таком виде работать неудобно. Поэтому адресам ячеек памяти, в которые записываются значения величин с целью их использования в программе, принято давать имя (**идентификатор**).

Имя переменной (идентификатор) всегда должно начинаться с латинской буквы либо знака подчеркивания, после чего могут следовать несколько латинских букв, цифры либо символ подчеркивания, записанные без пробелов.

Имя переменной не должно совпадать с именем программы.

Таким образом, переменные и константы — это именованные ячейки памяти, в которые записываются данные (например, числа). Если изобразить ячейки в виде прямоугольников, то это может выглядеть так:



Здесь в переменную с именем *A* записано число 10, а в переменную с именем *B* — число 6. В процессе выполнения программы значение каждой переменной может изменяться — на то она и переменная!

## 10.2. Типы данных

Каждая переменная, кроме имени и значения, имеет тип. **Тип переменной** определяет вид данных, содержащихся в переменной, набор операций, которые можно выполнять над этими данными, и множество допустимых значений.

В языке программирования Pascal существует несколько типов переменных, но в этой главе мы познакомимся только с двумя типами числовых данных:

- целым типом **Integer** (описывает диапазон значений целых чисел от  $-2\ 147\ 483\ 648$  до  $+2\ 147\ 483\ 647$ ; в памяти компьютера занимает 4 байт);
- вещественным типом **Real** (описывает диапазон значений дробных чисел (десятичных дробей), которые могут включать до 16 десятичных цифр; в памяти компьютера занимает 8 байт).

Переменные, используемые в программе, должны быть описаны в разделе описаний, расположеннном сразу после заголовка программы. **Описать переменную** — это значит указать после зарезервированного слова **Var** имя и тип переменной.

Например:

```
Var a: integer;
    Var a,b: real;
```

- ?
1. Что называют переменной?
  2. Каковы правила записи имен переменных?
  3. Какие типы числовых данных Вам известны?
  4. Что значит описать переменную?

## § 11. Организация ввода и вывода данных

При составлении программ можно выделить следующие этапы:

- ввод исходных данных;
- обработка данных;
- вывод данных (получение результата).

Для организации ввода и вывода данных необходимо научиться писать программы, которые позволяют вести диалог между компьютером и человеком (пользователем).

Мы уже знаем, что вывести сообщение на экран можно с помощью команды вывода `Write`. Это одна из наиболее часто используемых команд, ведь программа должна не только организовать диалог между пользователем и компьютером, но и сообщить пользователю о результатах своей работы, т. е. вывести данные на экран монитора.

Итак, информацию (текст) на экран монитора можно выводить с помощью команды `Write`. Например, так:

```
Program privet_1;
Begin
```

```
    Write ('Привет! Я компьютер, который может все!');
```

```
End.
```

Здесь команда (процедура) вывода `Write` выводит на экран все, что написано между апострофами в скобках в одну строку. После выполнения этой программы на экране монитора появится сообщение:

Привет! Я компьютер, который может все!

Программа `privet _ 2`, в которой уже две команды вывода `Write`, выдаст такой же результат.

```
Program privet _ 2;      {Заголовок программы}
Begin                  {Начало программы}
    Write ('Привет!');
    Write ('Я компьютер, который может все!');
End.                  {Конец программы}
```

Существует другая разновидность команды вывода — `Writeln`, которая действует так же, как команда `Write`, но после вывода информации переводит курсор на новую строку.

Так, программа `privet _ 3`

```
Program privet _ 3;
Begin
    Writeln ('Привет!');
    Write ('Я компьютер, который может все!');
End.
```

выведет текст в две строки:

Привет! Я компьютер, который может все!
--

Мы рассмотрели случаи использования команд вывода `Write` и `Writeln`, когда на экран монитора выводится текст. Однако компьютер может не только писать текст на экране, но и выполнять арифметические операции.

Познакомимся с новыми возможностями команды `Write` на примере программы `summa`.

```
Program summa;
Begin
    Write (2+3);      {Вычисление суммы и вывод
                      результата на экран}
End.
```

После выполнения программы summa на экране появится число 5. Так работает команда Write, если арифметическое выражение записано в скобках без апострофов. Если бы мы записали Write ('2+3'), на экране монитора был бы выведен текст: 2+3.

Вводить данные в компьютер можно с помощью команды (процедуры) ввода Read (либо Readln), которая записывается аналогично команде вывода, за исключением того, что в круглых скобках через запятую перечисляются имена переменных, например: Read(a,b) либо Readln(a1,a2,...,aN).

Эта команда заносит данные (числа) в переменные. При этом значения переменных зависят от того, какая информация набирается на клавиатуре во время выполнения программы. Вводимые с клавиатуры данные отображаются на экране монитора.

Команда Readln отличается от команды Read тем, что после ввода необходимого количества данных осуществляется переход курсора на следующую строку.

Рассмотрим на примерах, как выполняется команда Read.

**Пример 1.** Написать программу вывода числа, введенного пользователем с клавиатуры по запросу.

Программа может быть такой:

```
Program vvod_a;
Var a: integer; {Описание переменной a
целого типа}
Begin
  Write ('Введите любое число ');
  Readln (a); {Запрос на ввод числа с
  клавиатуры и занесение его в переменную a}
  Write ('Ваше число - ', a);
End.
```

Результат выполнения программы в окне вывода при  $a = 5$  будет следующим:

```
Ведите любое число 5
Ваше число — 5
```

При выполнении программы команда Read (либо Readln) велит компьютеру останавливаться и ждать, пока пользователь не введет какое-то число для занесения его в соответствующую переменную. Если, например, ввели число 5, то именно оно становится значением переменной  $a$  в памяти компьютера.

**Пример 2.** Написать программу, в которой осуществляется ввод числа с клавиатуры, умножение его на 2 и вывод результата на экран монитора.

Программа может быть такой:

```
Program vvod_x;
Var x: integer;
Begin
  Write ('Введите число ');
  Readln (x);
  Write ('Если вы умножите это число на
два, то получится ', 2*x);
End.
```

Результат выполнения программы в окне вывода при  $x = 25$  будет следующим:

```
Ведите число 25
```

```
Если вы умножите это число на два, то
получится 50
```

Выполним программу еще раз и введем в ответ на запрос значение 5. На экране появится число 10. Если выполнить программу еще раз и ввести число 7, на экране появится число 14. Каждый раз при выполнении программы значение переменной  $x$  будет изменяться.

**Пример 3.** Написать программу вычисления суммы и разности двух данных целых чисел с организацией ввода — вывода данных с помощью диалога между пользователем и компьютером.

Программа может быть такой:

```
Program summa2;           {Заголовок программы}
Var a,b,s: integer;      {Описание переменных
                           целого типа}
Begin
  Writeln ('Введите два числа');
  {Запрос на ввод исходных данных}
  Readln(a,b); {Ввод с клавиатуры чисел a и b}
  Writeln ('Сумма=', a+b); {Вывод результата}
  Writeln ('Разность=', a-b); {Вывод результата}
End.
```

Во время выполнения программы после сообщения Введите два числа введем с клавиатуры значения переменных  $a$  и  $b$ , например 6 и 8. Сделать это можно двумя способами: набрать числа в одной строке через пробел и нажать клавишу Enter либо каждый раз при вводе очередного числа нажимать клавишу Enter (тогда на экране каждое вводимое число будет занимать новую строку).

В результате выполнения программы summa2 переменная  $a$  получит значение 6, а переменная  $b$  — значение 8.

В окне вывода мы увидим:

Введите два числа

6 8

Сумма = 14

Разность = -2

- ? 1. Какие команды используются для ввода данных?  
2. Какие команды используются для вывода данных?  
3. Чем отличается команда Write от Writeln?  
4. Чем отличается команда Read от Readln?

### Упражнения

1. Что будет выведено на экран в результате выполнения следующей программы:

```
Program z1;
```

```
Begin
```

```
    Writeln ('Привет, это я, умный компьютер!');
```

```
    Writeln ('Я знаю, что 6*5=', 6*5);
```

```
End.
```

2. Составьте программу, в которой осуществляется ввод целого числа с клавиатуры и вывод на экран монитора значения этого числа, увеличенного в 5 раз.

3. Составьте программу, в которой осуществляется ввод целого числа с клавиатуры и вывод на экран монитора значения этого числа, увеличенного на 10.

4. Составьте программу, в которой осуществляется ввод трех целых чисел с клавиатуры и вывод на экран монитора значения их суммы.

5. Составьте программу, в которой осуществляется ввод двух целых чисел с клавиатуры и вывод на экран монитора значения их произведения.

## § 12. Команда присваивания. Арифметические операции и выражения. Стандартные функции

### 12.1. Команда присваивания

Мы уже знаем, что данные можно записывать в переменные, которые бывают целыми либо вещественными. Над переменными целого и вещественного типов можно выполнять арифметические операции: сложение, вычитание, умножение, деление.

Многие действия с числовыми данными выполняются путем вызова встроенных в язык программирования функций. Такие функции называют **стандартными**. Например, вычислить квадрат величины  $x$  любого типа можно с помощью функции  $\text{sqr}(x)$ , а модуль  $x$  — с помощью функции  $\text{abs}(x)$ .

Основная часть любой программы состоит из команд, которые обрабатывают данные. Одной из таких команд является **команда присваивания**. С помощью команды присваивания можно задавать значения переменных (вводить исходные данные), вычислять значения арифметических выражений и результат записывать в переменные. Рассмотрим на примере, как используется эта команда.

**Пример 1.** Написать программу вычисления суммы двух целых чисел.

Программа может быть такой:

```
Program summa_2; {Заголовок программы}
Var a,b,s:integer;
Begin
  a:=10;           {Команда присваивания
                     записывает в переменную a число 10}
  b:=6;           {Команда присваивания
                     записывает в переменную b число 6}
```

```

s:=a+b;           {Команда присваивания
                    вычисляет сумму значений переменных
                    a и b и записывает результат в пе-
                    ременную s}
Writeln(s);       {Вывод значения пере-
                    менной s на экран}
End.           {Конец программы}

```

После выполнения программы summa \_ 2 в окне вывода появится число 16.

В команде присваивания слева от «:=» пишется имя переменной, а справа может быть число, переменная либо выражение. В общем виде команду присваивания можно представить так:

<Имя переменной>:=<выражение>;

Здесь <выражение> задает порядок выполнения действий над данными и может включать в себя переменные и константы всех типов, знаки операций, стандартные функции, круглые скобки (например,  $(a+b)^2$ ). В простейшем случае выражение может состоять из одной переменной либо константы (например,  $b:=5$ ;  $a:=b$ ).

Запись выражений в программах имеет свои синтаксические правила. Например, невозможно записывать выражения в виде обыкновенных дробей. Запись  $4ac$  компьютер тоже не поймет, операцию умножения опускать нельзя: следует писать  $4*a*c$ .

Выполнение каждой операции в выражениях происходит с учетом ее приоритета. Порядок выполнения операций следующий:

- Выражения в скобках вычисляются в первую очередь.
- После вычисления значений выражений в скобках вычисляются функции.

- После функций выполняется умножение и деление (они имеют одинаковый приоритет и выполняются в порядке их следования слева направо).
- Далее выполняется сложение и вычитание в порядке их следования.

Рассмотрим еще один пример, демонстрирующий работу команды присваивания.

Пример 2. Написать программу, в которой данное число увеличивается сначала в 2 раза, а затем на единицу.

Программа может быть такой:

```
Program izmen_a;
Var a:integer;
Begin
  a:=5;           {Переменной а присваивается значение 5 – исходные данные}
  Writeln('a=',a); {Вывод на экран монитора исходного значения переменной а}
  a:=2*a; {Значение переменной а увеличивается в 2 раза}
  Writeln('a=',a); {Вывод на экран монитора промежуточного значения переменной а}
  a:=a+1; {Значение переменной а увеличивается на 1}
  Writeln('a=',a); {Вывод на экран монитора значения переменной а – результат}
End.
```

В результате выполнения программы `izmen_a` в окне вывода появится следующая информация:

```
a=5
a=10
a=11
```

! В результате выполнения команды присваивания в переменную помещается новое значение. Предыдущее значение переменной стирается.

## 12.2. Работа с переменными целого типа

Над переменными целого типа можно выполнять следующие арифметические операции:

- + (сложение);
- (вычитание);
- \* (умножение);
- `div` (целочисленное деление);
- `mod` (деление по модулю).

**Целочисленное деление (`div`)** отличается от обычной операции деления тем, что делит число нацело с отбрасыванием остатка.

Результат целочисленного деления равен нулю, если делимое меньше делителя. Например:

Выражение	Результат
<code>5 div 7</code>	0
<code>11 div 5</code>	2
<code>10 div 3</code>	3

**Деление по модулю (`mod`)** вычисляет остаток, полученный при выполнении целочисленного деления. Например:

Выражение	Результат
<code>11 mod 5</code>	1
<code>14 mod 3</code>	2
<code>5 mod 7</code>	5

Рассмотрим на примере, как используются арифметические операции `div` и `mod`.

Пример 3. Написать программу вывода целой части и остатка от деления  $a$  на  $b$ .

Программа может быть такой:

```
Program Cel_operac;
Var a,b,c:integer;
Begin
  a:=13;           {a и b – исходные данные}
  b:=5;
  c:=a div b;     {Вычисление целой части
                     от деления a на b}
  Writeln(a,' div ',b,'=',c);
  {Вывод результата}
  c:=a mod b;     {Вычисление остатка от
                     деления a на b}
  Writeln (a,' mod ',b,'=',c);
  {Вывод результата}
End.
```

В окне вывода в результате выполнения программы Cel\_operac появится следующая информация:

```
13 div 5=2
13 mod 5=3
```

Целочисленные переменные можно обрабатывать с помощью стандартных функций `abs(x)` и `sqr(x)`, где аргумент  $x$  — это переменная либо выражение целого типа.

- Функция `abs(x)` вычисляет абсолютное значение переменной  $x$ .
- Функция `sqr(x)` возводит в квадрат значение переменной  $x$ .

! Аргументы функции всегда пишутся в скобках. В качестве аргументов могут быть выражения, например: `abs (b*b-4*a*c)`.

**Пример 4.** Написать программу, демонстрирующую обработку целых чисел с помощью стандартных функций.

Программа может быть такой:

```
Program func _ int;
Var a,b,c:integer;
Begin
  a:=-9;           {Исходные данные}
  b:=abs(a);      {Функция abs(a) вычисляет
                    модуль a и присваивает его перемен-
                    ной b}
  Writeln('b=',b); {Вывод на экран монитора
                    значения переменной b}
  c:=sqr(b);       {Функция sqr(b) возводит
                    в квадрат значение переменной b и
                    присваивает его переменной c}
  Writeln('c=',c); {Вывод на экран монитора
                    значения переменной c}
End.
```

Результат выполнения программы `func _ int` в окне вывода:

b=9
c=81

### 12.3. Работа с переменными вещественного типа

Над переменными вещественного типа можно выполнять арифметические операции сложения, вычитания, умножения, обычного деления (/), а также обрабатывать их с помощью стандартных функций

`abs(x)`, `sqr(x)`, `trunc(x)`, `round(x)`, `frac(x)`, где аргумент  $x$  — это переменная либо выражение вещественного типа.

- Функция `trunc(x)` вычисляет целую часть вещественного числа  $x$ .
- Функция `round(x)` округляет вещественное число  $x$  до целого.
- Функция `frac(x)` вычисляет дробную часть вещественного числа  $x$ .
- Функция `abs(x)` вычисляет модуль числа  $x$ .
- Функция `sqr(x)` вычисляет квадрат числа  $x$ .

Арифметические операции сложения, вычитания, умножения и стандартные функции `abs(x)` и `sqr(x)` над переменными вещественного типа выполняются так же, как над переменными целого типа.

Рассмотрим на примерах, как используются арифметические операции обычного деления (`/`) и стандартные функции `trunc(x)`, `round(x)`.

Пример 5. Написать программу деления вещественной переменной  $a$  на вещественную переменную  $b$ .

Программа может быть такой:

```
Program real_operac;
Var a,b,c:real; {Описание вещественных переменных}
Begin
  a:=15.5; {Присваивание значения переменной a}
  b:=3.2; {Присваивание значения переменной b}
  c:=a/b;{/ – операция деления}
  Writeln ('a:b=',c);
End.
```

Результат выполнения программы `real _ operac` в окне вывода:

```
a:=b=4.84375
```

- ! В программировании целая часть десятичной дроби отделяется от дробной части точкой.  
На нуль делить нельзя.

## 12.4. Преобразования типов числовых данных

В программе могут одновременно использоватьсь переменные разных типов. В этом случае важно знать, как их преобразовывать для совместного использования. Так, в переменные вещественного типа можно записывать значения целочисленных переменных. А вот в переменные целого типа нельзя записывать значения вещественных переменных.

Чтобы поместить число типа `real` в переменную типа `integer`, нужно отбросить дробную часть с помощью функции `trunc(x)` или `round(x)`.

Рассмотрим на примерах, как это делается.

**Пример 6.** Написать программу, демонстрирующую преобразование переменной вещественного типа в переменную целого типа и наоборот.

Программа может быть такой:

```
Program int _ real;
Var c,d:integer; {Описание переменных целого типа}
      a,b:real; {Описание переменных вещественного типа}
Begin
      c:=8;           {Присваивание}
      a:=5.7;         {значения исходным данным}
```

```

b:=c;           {В переменную веществен-
    ного типа b записывается значение
    целочисленной переменной c}
Writeln ('b=',b);
Writeln ('a=',a);
c:=trunc(a);   {Выделяется целая часть
    числа a}
Writeln ('trunc(5.7)=',c);
d:=round(a);   {Число a округляется до
    ближайшего целого и помещается в
    переменную d}
Writeln('round(5.7)=',d);
End.

```

Результат работы программы int \_ real в окне вывода:

```

b=8
a=5.7
trunc(5.7)=5
round(5.7)=6

```

**Пример 7.** Написать программу, демонстрирующую действия над числовыми данными разных типов и поясняющую тип результата.

Программа может быть такой:

```

Program int _ real1;
Var c,d,x,y:integer;
      a,b:real;
Begin
  c:=10;           {c, a, d – исходные данные}
  a:=5.7;
  d:=3;
  b:=a+c;         {Если одно из слагаемых
    вещественное, то и результат (пере-
    менная b) будет вещественным}

```

```

Writeln ('b=',b);
y:=frac(b); {Результат (переменная у) вычис-
ления дробной части всегда будет целым}
Writeln ('frac(b)=',y);
a:=c/d; {Результат (переменная а) де-
ления всегда будет вещественным}
Writeln('a=c/d=',a);
d:= c div 4; {При делении нацело
делимое, делитель и результат всегда
будут целыми}
Writeln(c,'div 4 =',d);
x:= trunc (a); {Результат (х) выделения
целой части в вещественном числе (а)
всегда будет целым}
Writeln('trunc(a)=',x);
End.

```

Результат выполнения программы int \_ real1 в окне вывода:

```

b=15.7
frac(b)=7
a=c/d=3.33333333333333
10 div 4 =2
Trunc(a)=3

```

Результат выполнения программы int \_ real1 показывает, что дробная часть частного может быть очень длинной либо с повторяющимся периодом. В этом случае можно укоротить дробную часть числа, задав формат вывода значения переменной. Для этого в команде Writeln указывают количество позиций для всего числа и количество позиций для дробной части. Например:

```
Writeln(a:6:2);
```

В этом случае 6 позиций отводится на все число, включая десятичную точку, и 2 позиции отводится на дробную часть числа.

- ? 1. Какие арифметические операции применяются в программировании?
- 2. Какие стандартные арифметические функции Вам известны?

### Упражнения

1. Определите тип результата в выражениях:

$$4 + \frac{2 \cdot 3}{3}; \quad 2 + \frac{5}{2};$$

$$4 + 3 \cdot 1,5; \quad (6 - 4) \cdot 2 + 5.$$

2. Найдите ошибки в следующих программах:

```
Program int _ r;
Var a,x:integer;
    b,c:real;
Begin
  a:=10;
  b:=5.2;
  x:=b div 2;
  c:= round(x);
End.
```

```
Program int _ r1;
Var a,c:integer;
    b,d:real;
Begin
  a:=25;
  b:=a/2;
  c:=sqr(a);
  d:=a mod 10;
End.
```

3. Напишите программу, которая вычисляет частное от целочисленного деления двух заданных целых чисел и выводит на экран монитора результат.

4. Дано целое число. Напишите программу, которая уменьшает это число в 2 раза и выводит на экран монитора результат.

5. Дано целое число  $c$ . Напишите программу, которая возводит в квадрат частное от целочисленного

деления числа  $c$  на 5 и выводит на экран монитора результат.

6. Дано целое число  $c$ . Напишите программу, которая возводит в квадрат остаток от целочисленного деления числа  $c$  на 10 и выводит на экран монитора результат.

7. Дано вещественное число  $x$ . Напишите программу, которая вычисляет целую часть числа  $x$ , округленную до ближайшего целого, и выводит на экран монитора результат.

8. Дано вещественное число  $x$ . Напишите программу, которая уменьшает в 2 раза целую часть числа  $x$  и выводит на экран монитора результат.

### § 13. Составление линейных программ для решения задач

Мы научились с помощью языка программирования выполнять арифметические действия на компьютере, организовывать диалог между компьютером и пользователем. Эти знания помогут нам составлять программы для решения различных задач с помощью компьютера.

**Задача 1.** Дано некоторое натуральное число, большее 10. Написать программу, которая выводит на экран монитора его последнюю цифру.

Эту задачу можно решить с помощью следующего алгоритма:

1. Ввести с клавиатуры натуральное число, большее 10.

2. Выделить последнюю цифру данного натурального числа.

3. Вывести выделенную цифру на экран монитора.

Программа, реализующая данный алгоритм, может выглядеть так:

```
Program cifr_p;
Var A,sp:integer;
Begin
  Write('Введите число, большее 10: ');
  Readln (a); {Ввод с клавиатуры исходных данных}
  sp:=a mod 10; {Выделение последней цифры числа — числа единиц}
  Writeln('Последняя цифра ',sp); {Вывод результата}
End.
```

Результат выполнения программы в окне вывода при  $a = 1245$ :

```
Введите число, большее 10: 1245
Последняя цифра 5
```

**Задача 2.** Петя купил в магазине  $n$  порций мороженого по цене 840 р. и  $k$  плиток шоколада по цене 2300 р. Сколько всего денег потратил Петя?

Эту задачу можно решить с помощью следующего алгоритма:

1. Ввести с клавиатуры количество купленных порций мороженого и количество плиток шоколада.
2. Найти значение выражения  $840 \cdot n + 2300 \cdot k$ .
3. Вывести на экран монитора результат вычисления — сумму потраченных денег.

Программа, реализующая данный алгоритм, может выглядеть так:

```
Program summa_deng;
Var n,k,sum:integer;
```

**Begin**

```

Write('Введите количество порций мороже-
ного: ');
Readln (n);           {Ввод с клавиатуры
исходных данных}
Write('Введите количество плиток шокола-
да: ');
Readln (k);           {Ввод с клавиатуры
исходных данных}
sum:=840*n+2300*k;
Writeln('Сумма денег = ',sum); {Вывод ре-
зульта}

```

**End.**

Результат выполнения программы в Окне вывода при  $n = 2$  и  $k = 3$ :

```

Введите количество порций мороженого: 2
Введите количество плиток шоколада: 3
Сумма денег = 8580

```

Выше были рассмотрены примеры задач с использованием целочисленных переменных. Следующие две задачи демонстрируют совместное использование целочисленных и вещественных переменных.

**Задача 3.** Даны три целых числа  $a$ ,  $b$ ,  $c$ . Составить программу вычисления выражения  $\frac{(b^2 + c^3)}{|a|}$ .

Эту задачу можно решить с помощью следующего алгоритма:

1. Ввести с клавиатуры три целых числа.
2. Найти значение выражения, используя соответствующие арифметические операции и встроенные стандартные функции.
3. Вывести на экран результат вычисления.

Программа, реализующая данный алгоритм, может выглядеть так:

```
Program vvl;
Var a,b,c:integer;
    V:real; {В переменную V поместим ре-
              зультат вычисления выражения}
Begin
    Write('a=');           {Ввод с клавиатуры}
    Readln (a);           {числа a}
    Write('b=');
    Readln (b);           {числа b}
    Write('c=');
    Readln (c);           {числа c}
    V:= (b*b+c*c)/abs(a);
    Writeln ('V=', V:4:2); {Вывод результата}
End.
```

Результат выполнения программы в окне вывода при  $a = -4$ ,  $b = 2$  и  $c = 3$  такой:

```
a=-4
b=2
c=3
v=7.75
```

**Задача 4.** Расстояние от города  $A$  до города  $B$  равно  $s$  км. Поезд проехал это расстояние за  $t$  ч. С какой скоростью двигался поезд ( $s$  и  $t$  — целые числа)?

Эту задачу можно решить с помощью следующего алгоритма:

1. Ввести с клавиатуры расстояние  $s$  и время  $t$ .
2. Вычислить скорость движения поезда:  $v = \frac{s}{t}$ .
3. Вывести на экран монитора результат.

Программа, реализующая данный алгоритм, может выглядеть так:

```
Program skor;
Var s,t:integer;
      v:real;
Begin
  Write('Расстояние = ');
  Readln (s);           {Ввод с клавиатуры
                        значения переменной s}
  Write('Время = ');
  Readln (t);           {Ввод с клавиатуры
                        значения переменной t}
  v:= s/t;              {Вычисление скорости}
  Writeln('v = ',v:4:2); {Вывод результата}
End.
```

Результат выполнения данной программы получите самостоятельно.

### Упражнения

1. Даны целые числа  $a$  и  $b$ . Напишите программу, которая находит значение выражения  $\frac{(a^2 + b)a}{b}$  и выводит на экран монитора результат.
2. Даны целые числа  $x$ ,  $y$ ,  $c$ . Напишите программу, которая находит значение выражения  $\frac{x+y}{2c}$  и выводит на экран монитора результат.
3. Дано целое число. Напишите программу, которая уменьшает на 2 модуль этого числа и выводит на экран монитора результат.
4. Дано вещественное число. Напишите программу, которая возводит в квадрат целую часть данного числа и выводит на экран монитора результат.

5. Дано двузначное целое число. Напишите программу, которая выводит на экран монитора его первую цифру.

6. Дано целое число. Напишите программу, которая выводит на экран последнюю цифру данного числа, затем возводит ее в квадрат и выводит на экран монитора результат.

7. Напишите программу, которая вычисляет расстояние, пройденное автомобилем за время  $t$  при движении со скоростью  $v \frac{\text{км}}{\text{ч}}$ , и выводит на экран монитора результат.

8. Дан рост трех друзей. Напишите программу, которая определяет их средний рост и выводит на экран монитора результат.

## § 14. Графические возможности языка программирования

Мы научились составлять программы, с помощью которых компьютер пишет на экране монитора, считает, решает задачи. В этом параграфе мы познакомимся с графическими возможностями языка программирования.

### 14.1. Графический режим работы

При работе в графическом режиме изображение на экране строится из точек — **пикселей**. Каждый пиксель (точка) характеризуется координатами и цветом.

Количество точек, с помощью которых на экране монитора можно построить изображение, называется его **разрешающей способностью (разрешением)**.

Разрешение монитора может задаваться выражением  $x \times y$ , где  $x$  — количество пикселей на экране

по горизонтали, а  $y$  — по вертикали. Например, известны следующие разрешения мониторов:

$320 \times 200$ ;  $640 \times 480$ ;  $800 \times 600$ ;  $1024 \times 768$ ;  $1280 \times 1024$  и т. д.

Даже при самом маленьком разрешении монитора изображение в графическом режиме формируется с помощью 64 000 ( $320 \cdot 200$ ) графических элементов. Из такого количества «строительных кирпичиков» можно построить довольно сложную картинку.

## 14.2. Возможности стандартной графической библиотеки

Изображение можно сформировать из простых геометрических фигур: точек, отрезков, прямоугольников, окружностей и т. д. Положение и размеры геометрического объекта задаются координатами его точек. Поскольку минимальным элементом изображения является пиксель, в качестве координат используют порядковые номера пикселей по горизонтали и вертикали. Началом отсчета является левый верхний угол экрана. Значения координат  $x$  отсчитываются слева направо, а зна-

чения координаты  $y$  — сверху вниз (рис. 3.4). Например, координаты точки  $(20, 50)$  означают, что от левого верхнего угла экрана надо отсчитать 20 пикселей по горизонтали вправо и 50 пикселей по вертикали вниз.

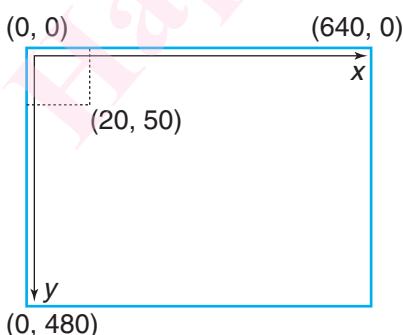


Рис. 3.4

! Координаты пикселя принимают только целочисленные значения.

Координаты пикселя принимают значения, ограниченные нулем и размерами графического окна.

Рисование различных геометрических фигур осуществляется с помощью специальных стандартных команд (процедур), которые хранятся в библиотечном модуле GraphAbc. Эту библиотеку подключают в разделе описаний с помощью зарезервированного слова **Uses** (подключить).

Рисование осуществляется в графическом окне. Возможность рисовать одновременно в нескольких окнах отсутствует. Рисовать можно с помощью пера и кисти.

Размеры графического окна можно задавать в модуле GraphAbc с помощью команды **SetWindowSize(X,Y)** (табл. 5 Приложения). По умолчанию окно занимает весь экран монитора.

Изучать графические возможности системы программирования Pascal ABC мы будем на конкретных примерах.

Пример 1. Написать программу, которая выведет на экран монитора черную точку на белом фоне.

```
Program tochka;
Uses GraphAbc;           {Подключение модуля
                           GraphAbc}
Begin
  SetWindowSize(640,480); {Установка размеров
                           графического окна}
```

```
SetPixel(100,120,clBlack); {Рисование точки с координатами (100,120) черным цветом}
```

**End.**

В данном примере мы познакомились с использованием команды SetPixel. В общем виде эта команда выглядит так:

```
SetPixel(x,y,color),
```

где x, y — координаты точки, color — ее цвет.

Стандартные цвета color задаются символьическими константами:

clBlack — черный, clPurple — фиолетовый, clWhite — белый, clMaroon — темно-красный, clRed — красный, clNavy — темно-синий, clGreen — зеленый, clBrown — коричневый, clBlue — синий, clSkyBlue — голубой, clYellow — желтый,	clCream — кремовый, clAqua — бирюзовый, clOlive — оливковый, clFuchsia — сиреневый, clTeal — сине-зеленый, clGray — серый, clLime — ярко-зеленый, clLtGray — светло-серый, clDkGray — темно-серый, clSilver — серебряный.
---	--

### 14.3. Рисование линий

С помощью модуля GraphAbc можно рисовать геометрические фигуры: линии, прямоугольники и окружности.

Линии можно рисовать с помощью команды Line, которая имеет общий вид:

```
Line(x1,y1,x2,y2),
```

где (x1,y1) и (x2,y2) — координаты начала и конца отрезка.

**Пример 2.** Написать программу рисования отрезка красного цвета, соединяющего две точки с координатами (120,150) и (300,100).

Программа может быть такой:

```
Program Linii;
Uses GraphABC;
Begin
  SetWindowSize(640,480);
  SetPenColor(clRed);      {Установка красного
                           цвета пера}
  Line(120,150,300,100);   {Рисование отрезка
                           красным цветом от точки с координатами
                           (120,150) до точки с координатами
                           300,100)}
End.
```

На рисунке 3.5 показано, что мы увидим на экране монитора после выполнения программы.

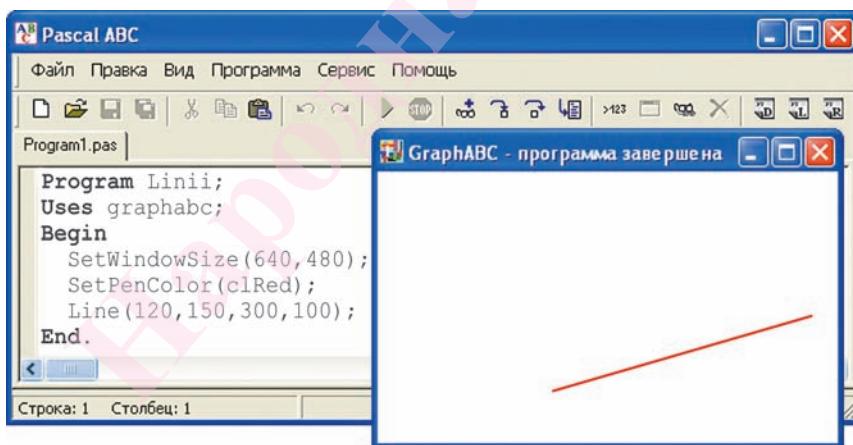


Рис. 3.5

**Пример 3.** Написать программу, которая линией определенной толщины нарисует треугольник, у которого все стороны разного цвета.

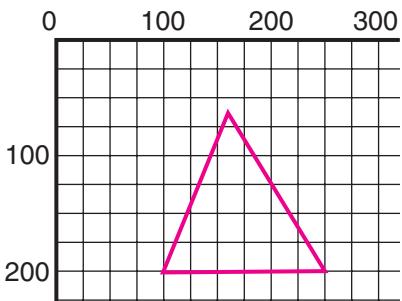


Рис. 3.6

С помощью тетради в клеточку определим координаты вершин треугольника на экране монитора (рис. 3.6).

Программа может быть такой:

```

Program treugolnik;
Uses GraphAbc;
Begin

    SetWindowSize(640,480);
    SetPenWidth(5);           {Установка толщины
        текущего пера. В скобках указывается количество
        пикселей, образующих толщину
        линии}
    SetPenColor(clRed);
        {Установка красного цвета пера}
    Line(100,200,170,70);     {Рисование отрезка
        красным цветом от точки с координатами
        (100,200) до точки с координатами
        (170,70)}
    SetPenColor(clGreen);   {Установка зеленого
        цвета пера}
    Line(170,70,250,200);    {Рисование отрезка
        зеленым цветом от точки с координатами
        (170,70) до точки с координатами
        (250,200)}
    SetPenColor(clBlue);    {Установка синего
        цвета пера}
    Line(250,200,100,200);   {Рисование отрезка
        синим цветом от точки с координатами
        (250,200) до точки с координатами
        (100,200)}

End.

```

На рисунке 3.7 показан результат выполнения программы `treugolnik`.

Рисуя линию, можно задавать ее размер, цвет, ширину (толщину) и стиль.

Для задания стиля линии в модуле GraphAbc существует команда

`SetPenStyle(Style),`

где `Style` — константы стилей пера (табл. 6, 7 Приложения). Стиль задает вид линии (сплошная, пунктирная, штрихпунктирная, штриховая).

Замкнутые фигуры можно закрашивать с помощью команды заливки

`FloodFill(x,y,color),`

где `(x,y)` — координата точки, которая обязательно должна попасть во внутреннюю область закрашиваемой фигуры, а `color` — цвет заливки.

Пример 4. Написать программу, которая нарисует треугольник и закрасит его.

Программа может быть такой:

```
Program treug_zakrash;
Uses GraphAbc;
Begin
  SetWindowSize(640,480);
  ClearWindow(clWhite); {Очистка графического окна белым цветом}
  SetPenWidth(3); {Установка толщины текущего пера}
  SetPenStyle(psSolid); {Установка стиля линии — сплошная линия}
  SetPenColor(clGreen); {Установка зеленого цвета пера}
```

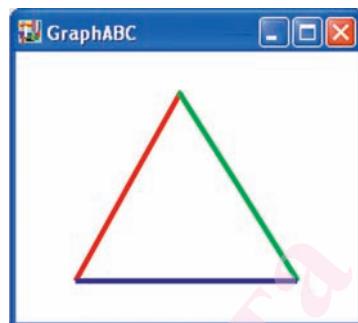


Рис. 3.7

```

Line(100,200,170,70);      {Рисование линии
                           зеленым цветом}
Line(170,70,250,200);
Line(250,200,100,200);
FloodFill(150,120,clRed); {Закрашивание тре-
                           угольника красным цветом}

```

**End.**

В результате выполнения программы на экране монитора в графическом окне появится красный треугольник с зеленым контуром (рис. 3.8).

! Закрашивать можно только замкнутые фигуры, контур которых нарисован одним цветом.

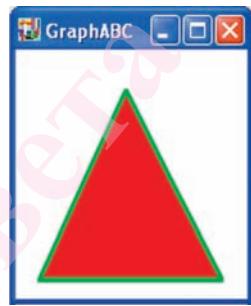


Рис. 3.8

#### 14.4. Рисование прямоугольников и окружностей

Прямоугольники и окружности можно рисовать с помощью команд `Rectangle(x1,y1,x2,y2)` и `Circle(x,y,r)` соответственно, описание которых приведено в таблице 8 Приложения.

Рассмотрим на примере применение данных команд.

**Пример 5.** Написать программу рисования прямоугольника и окружности.

Программа может быть такой:

```

Program geometr;
Uses GraphAbc;
Begin
  SetWindowSize(640,480);
  SetPenColor(clBlue);      {Установка синего
                           цвета пера для рисования контура
                           прямоугольника}

```

```
SetPenWidth(6);           {Установка
    толщины пера}
Rectangle(50,50,250,150); {Рисование
    прямоугольника, заданного координатами
    (50, 50) и (250, 150) противоположных
    вершин}
SetPenColor(clRed);       {Установка
    красного цвета пера для рисования
    контура окружности}
Circle(350,100,60);      {Рисование
    окружности с центром в точке с
    координатами (350,100) и радиусом 60}
End.
```

Результат выполнения программы приведен на рисунке 3.9.



Рис. 3.9

С помощью команд SetTextColor, SetFontSize, SetBrushColor, TextOut, общий вид которых приведен в таблицах 8—11 Приложения, рисунки можно подписывать.

**Пример 6.** Написать программу, которая нарисует прямоугольник и окружность, закрасит и подпишет их.

Программа может выглядеть так:

```
Program geometr3;
Uses GraphAbc;
```

**Begin**

```
SetWindowSize(640,480);
ClearWindow(clYellow); {Установка желтого
цвета фона}
SetPenColor(clTeal); {Установка сине-
зеленого цвета пера}
SetPenWidth(5); {Установка толщины линии}
SetBrushColor(clOlive); {Установка оливко-
вого цвета кисти}
Rectangle(100,100,300,200); {Рисование за-
крашенного оливковым цветом прямо-
угольника}
SetBrushColor(clBlue); {Установка синего
цвета кисти}
Circle(400,150,50); {Рисование закрашенного
синим цветом круга}
SetFontStyle(fsbold); {Установка стиля
шрифта}
FontSize(15); {Установка размера
шрифта в пикселях}
SetBrushColor(clWhite); {Установка белого
цвета кисти}
SetFontColor(clOlive); {Установка оливко-
вого цвета шрифта}
TextOut(100,220,'Прямоугольник'); {Размеще-
ние надписи в прямоугольнике с коор-
динатами левого верхнего угла
(100,220)}
SetFontColor(clBlue); {Установка синего
цвета шрифта}
TextOut(380,220,'Круг'); {Размещение над-
писи в прямоугольнике с координатами
левого верхнего угла (380,220)}
```

**End.**



Рис. 3.10

На рисунке 3.10 показан результат выполнения программы.

Из приведенного примера видно, что рисовать можно на «холсте» определенного цвета, который задается с помощью команды очистки графического окна заливкой ClearWindow(color) (табл. 5 Приложения). Закрасить прямоугольник и круг можно, используя команду установки цвета кисти SetBrushColor (color) (табл. 9, 10 Приложения).

**Пример 7.** Используя рассмотренные выше команды и примеры, написать программу, которая с помощью простейших геометрических фигур нарисует картинку, изображенную на рисунке 3.11.

Программа, в результате выполнения которой получится такая картинка, имеет вид:

```
Program skvoreshnik;
Uses GraphAbc;
Begin
  SetWindowSize(640,480);
```

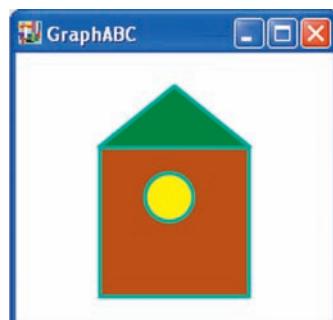


Рис. 3.11

```

ClearWindow(clWhite);    {Установка белого
    цвета фона}
SetPenColor(clTeal);    {Установка сине-
    зеленого цвета пера}
SetPenWidth(5);          {Установка толщины
    линии}
Rectangle(100,150,250,300); {Рисование
    прямоугольника}
SetBrushColor(clYellow); {Установка желто-
    го цвета кисти для закраски круга}
Circle(170,200,25);    {Рисование закрашенно-
    го желтым цветом круга}
FloodFill(150,160,clBrown); {Закрашивание
    коричневым цветом прямоугольника}
Line (100,150,175,90); {Рисование крыши}
Line (175,90,250,150);
FloodFill(175,100,clGreen); {Закрашивание
    треугольника (крыши) зеленым
    цветом}

```

**End.**

В модуле GraphAbc имеются и другие команды рисования геометрических фигур. Их Вы можете изучить самостоятельно с помощью Приложения.

- ?
1. Из каких геометрических фигур можно создать изображение?
  2. Что является минимальным элементом графического изображения?
  3. Как задаются координаты точки на экране монитора?
  4. Какая команда рисует линию?
  5. Какая команда рисует прямоугольник?
  6. Какая команда рисует окружность?

## Упражнения

**1.** Что будет выведено на экран монитора в результате выполнения следующих программ:

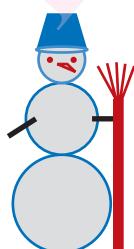
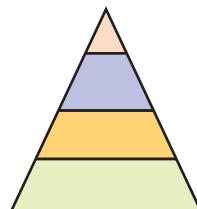
```
Program grafika1;
Uses GraphAbc;
Begin
  SetPenWidth (10);
  SetPenColor (clRed);
  Line (100,100,270,90)
End.
```

```
Program grafika2;
Uses GraphAbc;
Begin
  SetPenWidth (8);
  SetPenColor (clBlue);
  Circle (200,150,50)
End.
```

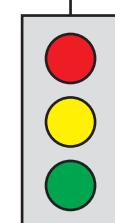
**2.** С помощью известных Вам графических команд напишите программу, которая изобразит на экране монитора:

- а) желтый квадрат на зеленом фоне;
- б) зеленый ромб на голубом фоне;
- в) избушку на курьих ножках;
- г) лодку с парусом.

**3.** Используя известные Вам команды рисования геометрических фигур, нарисуйте картинки:



Красный свет — дороги нет



## Глава 4

# КОМПЬЮТЕРНЫЕ ПРЕЗЕНТАЦИИ

### § 15. Понятие компьютерной презентации. Программа PowerPoint

Презентацией (от лат. *praesentatio* — предъявление) называют представление чего-либо нового, недавно созданного, например автомобиля, книги, кинофильма, песни, телепрограммы. Презентация как способ представления информации широко используется для рекламы товаров, пропаганды научных, технических и спортивных достижений. Формы презентаций весьма разнообразны: демонстрация моделей одежды, театрализованное представление, лекция, доклад и т. д. Сегодня для подготовки и проведения презентаций активно используются компьютеры.

Под компьютерной презентацией понимают созданный на компьютере и предназначенный для показа электронный документ, содержащий тексты, рисунки, таблицы, диаграммы. Он может также содержать эффекты анимации, звуковое сопровождение и видеофрагменты.

Для подготовки компьютерных презентаций можно использовать текстовые и графические редакторы. Однако наиболее мощными инструментами создания и демонстрации презентаций являются специальные презентационные программы, например Microsoft PowerPoint, Open Office Impress.

Программа PowerPoint, предназначенная для подготовки и демонстрации электронных презентаций, является офисным приложением Microsoft Office.

Презентация в PowerPoint представляет собой последовательность сменяющих друг друга слайдов. Понятие **слайд** используется для обозначения каждой страницы презентации. Слайды могут сменяться

автоматически через заданные промежутки времени или вручную по нажатию клавиши на клавиатуре (щелчку мыши). В простейшем случае презентация может содержать всего один слайд (например, объявление, открытка или рекламный плакат).

Подготовленные слайды презентации могут быть распечатаны на бумаге, прозрачной пленке или продемонстрированы на компьютере.

Основные возможности программы PowerPoint:

- создание слайдов презентации, содержащих тексты, рисунки, таблицы, диаграммы, аудио- и видеофрагменты;
- задание порядка следования слайдов, использование эффектов анимации;
- сохранение и демонстрация презентации.

Программу PowerPoint можно запустить, выполнив команды: Пуск → Программы → Microsoft Office → Microsoft PowerPoint или щелкнув мышью

на ярлыке  на Рабочем столе.

После запуска программы открывается окно, основные элементы которого представлены на рисунке 4.1. В Строке заголовка отображается имя создаваемой или редактируемой презентации. Стока меню содержит команды работы с основным окном PowerPoint и слайдами презентации. В Строке состояния отображается номер текущего слайда, используемое оформление и язык набора текста. Панели инструментов Стандартная, Форматирование и Рисование по своим возможностям напоминают аналогичные панели текстового редактора Word.

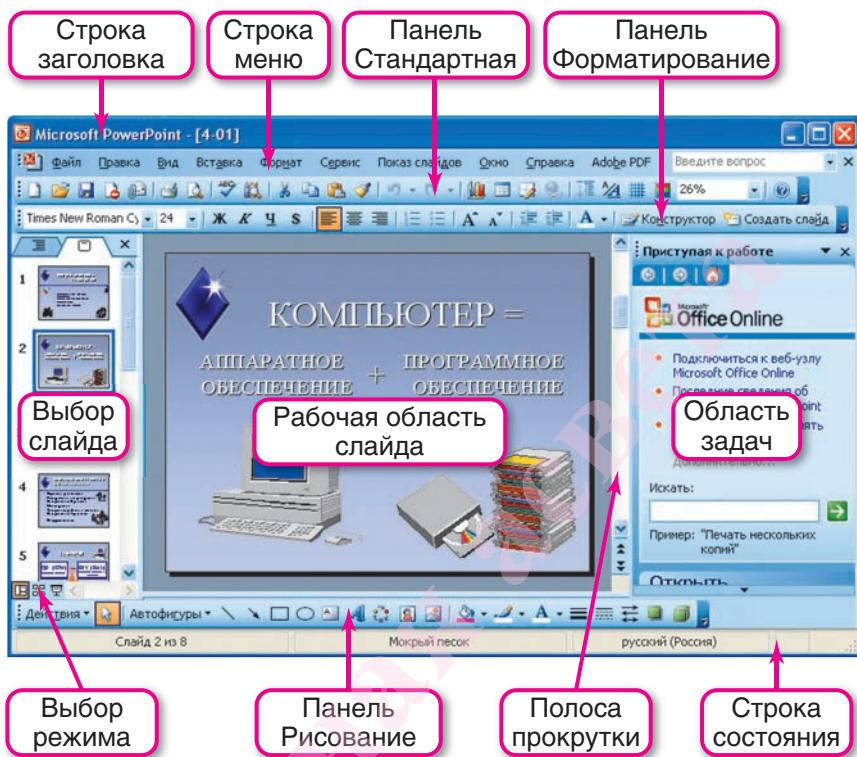


Рис. 4.1

Программа PowerPoint позволяет работать с электронной презентацией в режимах **Обычный**, **Сортировщик слайдов**, **Показ слайдов**. Режим работы выбирается с помощью меню **Вид** или кнопок, расположенных в нижней части окна программы (см. рис. 4.1).

**Обычный режим** предназначен для создания и редактирования слайдов. Основную часть экрана занимает **рабочая область слайда** (см. рис. 4.1). Она обеспечивает размещение на слайде текста, изображений, таблиц, диаграмм, видеофрагментов и других объектов. Операции по работе с текстом и изображения-

ми похожи на аналогичные действия в текстовом или графическом редакторе. Для удобства выбора различных операций, например разметки слайдов, настройки анимации, используется **область задач**.

С помощью области **Выбор слайда** можно выбрать необходимый для редактирования или просмотра слайд. Можно удалить выбранные слайды (нажатием клавиши Del) или изменить их порядок перетаскиванием с нажатой левой кнопкой мыши. Для последовательного перемещения между слайдами презентации используются **полосы прокрутки**.

В режиме **Сортировщик слайдов**  экран заполнен миниатюрными изображениями слайдов презентации (рис. 4.2). Содержание слайдов в этом режиме изменять нельзя, но можно удалять и копировать слайды, а также изменять порядок их следования.

Режим **Показ слайдов**  позволяет демонстрировать презентацию, последовательно разворачивая слайды на весь экран, начиная с текущего.

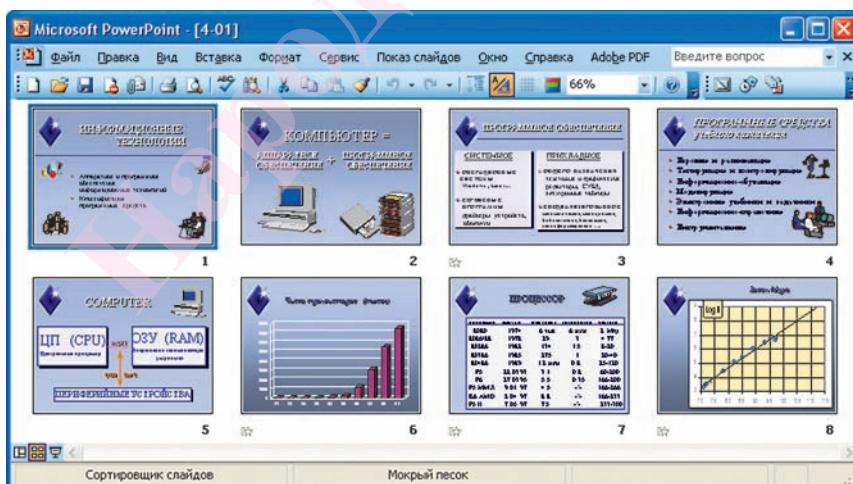


Рис. 4.2

Пример. Открыть предложенную учителем презентацию. Познакомиться с основными элементами интерфейса и режимами работы.

С помощью меню **Файл** → **Открыть** или комбинации клавиш **Ctrl + O** вызовем диалоговое окно **Открытие документа** и выберем предложенный файл, например **primer.ppt**.

В режиме **Обычный** просмотрим содержание всех слайдов, последовательно выбирая их в области **Выбор слайда** или с помощью полосы прокрутки. Поменяем местами слайды 4 и 5, перетащив один из них с нажатой левой кнопкой мыши. Удалим слайды 7 и 8, выделив их и нажав клавишу **Del**.

Переключимся в режим **Сортировщик слайдов**. Поменяем местами слайды 3 и 4. Удалим слайды 5 и 6.

Выберем 3-й слайд и щелкнем мышью на кнопке . Выбранный слайд развернется на весь экран и будет оставаться на нем до тех пор, пока не поступит команда смены слайда (щелчок мыши или нажатие клавиши на клавиатуре) либо не закончится установленное время.

Для показа всей презентации выберем первый слайд и щелкнем мышью на кнопке .

Начать показ всей презентации можно также с помощью меню **Показ слайдов** → **Начать показ** или клавиши **F5**.

- ? 1. Что понимается под компьютерной презентацией?  
2. Каково назначение и основные возможности программы PowerPoint?  
3. С помощью каких действий можно начать показ презентации?

## Упражнение

Откройте и продемонстрируйте предлагаемые учителем презентации. Поменяйте местами четные и нечетные слайды. Удалите указанные учителем слайды.

### § 16. Создание презентации

Создание компьютерной презентации начинают с планирования ее структуры, т. е. определения количества слайдов и последовательности их показа. Следует заранее подготовить материалы для размещения на слайдах. После этого можно приступить непосредственно к работе. Существует несколько способов создания презентаций. В MS Office2003 удобно использовать область задач.

Пример. Создать презентацию «Неделя информатики» из двух слайдов с текстом.

Откроем редактор презентаций PowerPoint. По умолчанию отображается область задач **Приступая к работе** (при отсутствии ее можно вызвать с помощью меню **Вид**).

Выбрав пункт **Создать презентацию**, перейдем в область задач **Создание презентации** (можно также использовать меню **Файл → Создать** или комбинацию клавиш **Ctrl + N**). Здесь предлагается выбрать способ создания: **Новая презентация**, **Из шаблона оформления**, **Из мастера автосодержания**, **Из имеющейся презентации** (рис. 4.3).

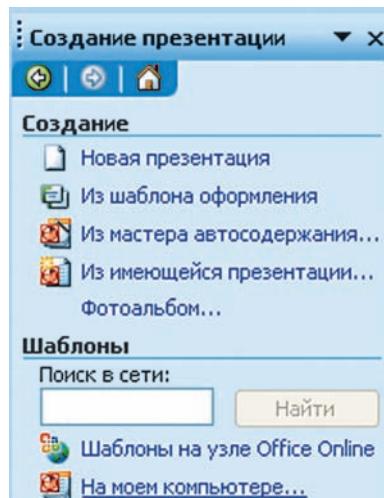


Рис. 4.3

Выберем способ **Новая презентация**. Откроется область задач **Разметка слайда**, в которой предлагаются макеты, содержащие ограниченные рамками области слайда для ввода текста (их называют *надписями*) и других объектов. Под **макетом слайда** понимают порядок размещения на нем различных элементов: заголовков, текстов, списков, рисунков, таблиц и др. Конечно, все объекты можно размещать на слайде по своему усмотрению. Однако нередко проще использовать подходящий готовый макет, наполняя его своим содержанием.

**1. Создание слайда.** Для первого слайда презентации по умолчанию предлагается макет **Титульный слайд**. Он содержит две прямоугольные рамки, предназначенные для размещения названия презентации, а также иной заглавной информации, например фамилий авторов (рис. 4.4, *a*).

Щелчком мыши выделим верхнюю рамку с подсказкой «Заголовок слайда» и введем текст **НЕДЕЛЯ ИНФОРМАТИКИ** (шрифт Arial, 60 пт, жирный, выравнивание по центру). Заметим, что текст выравнивается относительно рамки, а не всего слайда. Для перехода на новую строку без создания абзаца используем комбинацию клавиш Shift + Enter.

Выделим нижнюю рамку с подсказкой «Подзаголовок слайда» и введем текст **Конкурсы и олимпиады, турниры и викторины** (шрифт Arial, 32 пт, нормальный вес).

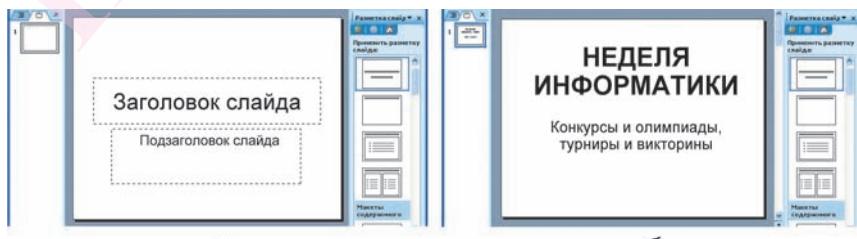


Рис. 4.4

мальный, выравнивание по центру). Подкорректируем положение надписей в соответствии с рисунком 4.4, б. Первый слайд создан, однако он выглядит не-привлекательно.

**2. Использование шаблона оформления.** Займемся оформлением слайда. Воспользуемся готовыми шаблонами оформления. Для этого отобразим область задач **Дизайн слайда**, в которой можно выбрать **Шаблоны оформления**, **Цветовые схемы**, а также **Эффекты анимации** (рис. 4.5).

**Шаблоном оформления** (или **темой**) называют набор элементов с подобранными параметрами: размеры и положение рамок надписей, шрифт и цвет текста, фоновые изображения и другие свойства. Шаблон может быть применен к одному или сразу ко всем слайдам презентации.

Подберем подходящий шаблон, последовательно выбирая его из предлагаемых. При этом может потребоваться коррекция положения надписей. На рисунке 4.6 приведены результаты применения шаблонов **Салют** (рис. 4.6, а), **Палитра** (рис. 4.6, б), **Пастель** (рис. 4.6, в, г). Заметим, что некоторых из этих шаблонов на Вашем компьютере может не оказаться. С программой PowerPoint поставляется лишь несколько шаблонов. В этом случае выберите другой

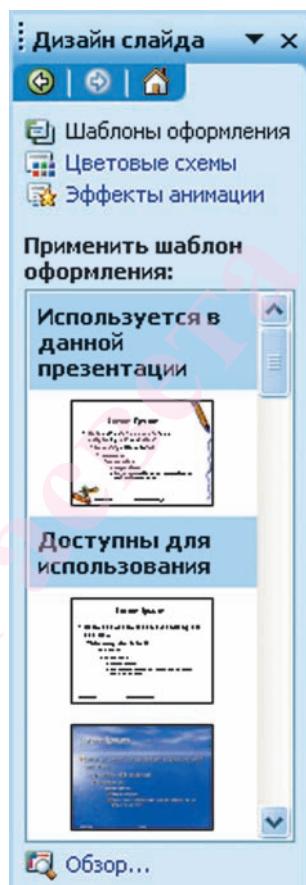


Рис. 4.5



Рис. 4.6

подходящий шаблон. При необходимости большое количество шаблонов можно скачать из сети Интернет или создать самостоятельно.

Для каждого шаблона оформления предлагается несколько вариантов сочетаний цветов, которые называются **цветовыми схемами**.

**3. Подбор цветовой схемы.** Откроем область задач **Цветовые схемы** (рис. 4.7). Попробуем применить некоторые из них к созданному слайду. Рисунок 4.6 иллюстрирует результаты применения (сравните рис. 4.6, в и г). Остановим свой выбор на шаблоне **Пастель** и предлагаемой по умолчанию цветовой схеме, показанной на рисунке 4.6, в.

Конечно, можно вручную задавать рисунок и цвет фона и каждой надписи, однако при этом приходится

самостоятельно подбирать сочетания цветов, что требует терпения и художественного вкуса. В любом случае полезно сначала попробовать готовые шаблоны и цветовые схемы и лишь на последнем этапе оформления корректировать их.

Создадим теперь второй слайд, используя тот же шаблон оформления. Для этого щелкнем мышью по значку **Создать слайд**  Панели инструментов (можно воспользоваться меню **Вставка** → **Создать слайд** или комбинацией клавиш **Ctrl + M**). Слайд должен содержать заголовок **Конкурсы** и их перечисление, поэтому в области задач **Разметка слайда** выберем макет **Заголовок и текст**. Заметим, что этот макет предлагается по умолчанию при создании второго слайда. Если нужно создать слайд с иной разметкой, выбирают более подходящий макет.

Введем и отформатируем тексты в соответствии с рисунком 4.8.

Созданную презентацию необходимо сохранить. Для этого выполним последовательность команд: **Файл** → **Сохранить как...** → введем имя, например **Инфо1** → **Сохранить**. Все слайды презентации хранятся в одном файле с расширением **.ppt**.

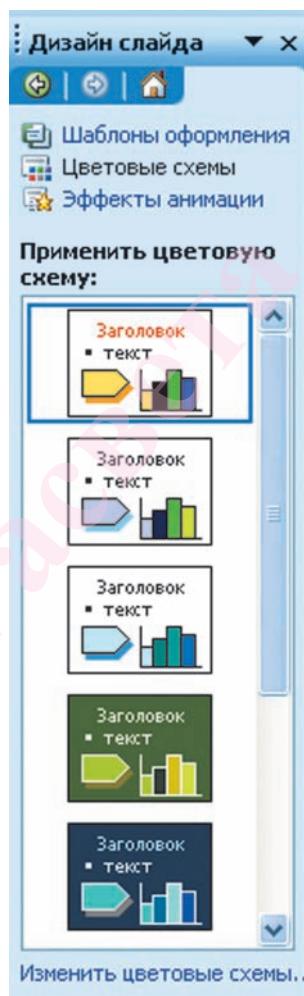


Рис. 4.7

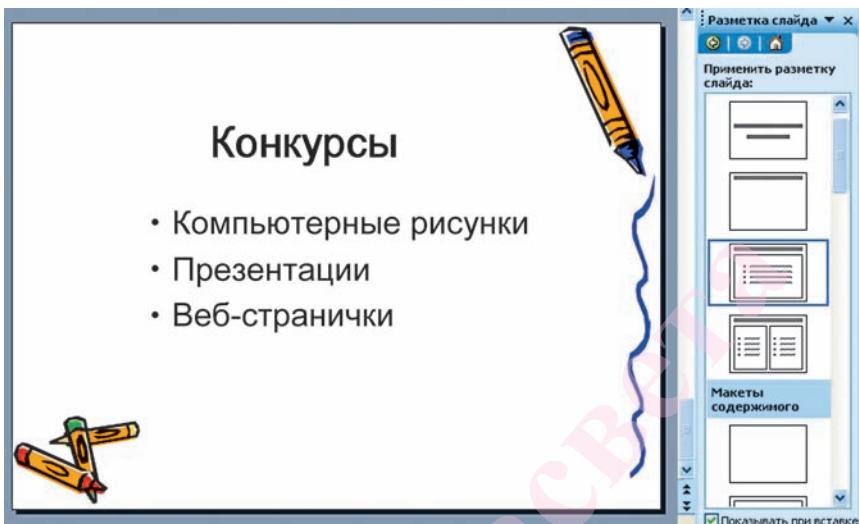


Рис. 4.8

Наконец продемонстрируем созданную презентацию. Для этого воспользуемся меню **Показ слайдов** → **Начать показ** или клавишей **F5**.

Презентацию можно распечатать на бумаге или прозрачной пленке с помощью меню **Файл** → **Печать**. В появившемся диалоговом окне надо указать количество слайдов на листе бумаги и порядок их размещения.

- ?
1. Для чего предназначены макеты слайдов?
  2. Что понимают под шаблоном оформления презентации?
  3. Что понимают под цветовой схемой?
  4. Какое расширение имеет файл с презентацией?

### *Упражнения*

1. Добавьте к созданной в примере презентации третий слайд, который содержит надпись **Победителей конкурсов ждут призы**. Подберите и примените к этому слайду подходящий шаблон оформления.

2. Создайте презентацию по одной из тем: «Моя школа», «Мой рабочий день», «Мои друзья», «Мое хобби».

## § 17. Вставка изображений. Использование коллекций

В презентации важную роль играют элементы оформления: рисунки, таблицы, графики, звук, видеофрагменты. Использование этих объектов делает презентацию более яркой и наглядной. Следует заранее подготовить материалы для размещения на слайдах, преобразовать их в цифровую форму и сохранить в файлах на компьютере (например, отсканировать, скачать из сети Интернет или переписать с цифрового фотоаппарата).

Программы Microsoft Office имеют встроенные коллекции картинок, звуков, видеофрагментов. Они называются **клипартами**. Эти картинки можно использовать для оформления презентации, например в качестве символов и эмблем.

Пример. Создать презентацию «Функциональные блоки компьютера».

Откроем редактор презентаций PowerPoint. В области задач **Создание презентации** выберем способ **Из шаблона оформления**. Откроется область задач **Дизайн слайда с шаблонами оформления**. Выберем подходящий шаблон, например **Эхо**.

На титульном слайде выделим верхнюю рамку с подсказкой «Заголовок слайда» и введем текст **ФУНКЦИОНАЛЬНЫЕ БЛОКИ КОМПЬЮТЕРА** (шрифт Arial, 44 пт, жирный, выравнивание по левому краю). Выделим нижнюю рамку с подсказкой «Подзаголовок слайда» и введем текст **в картинках**

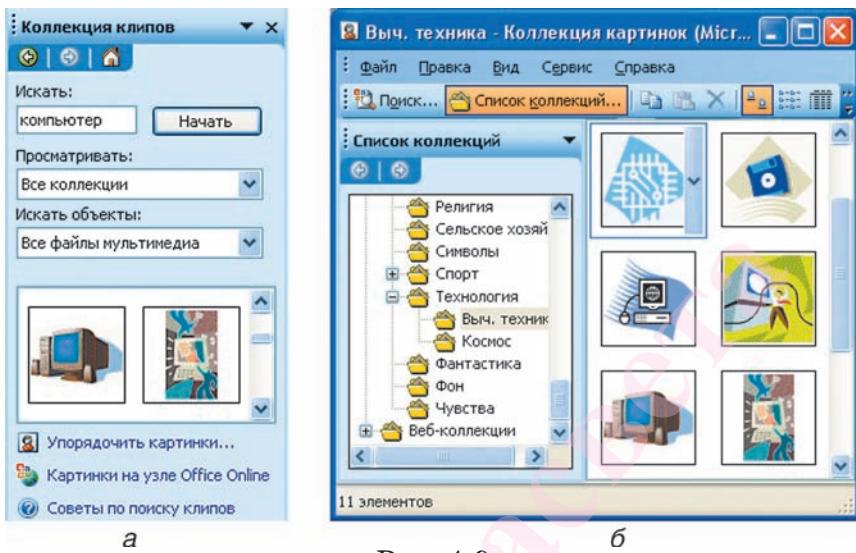


Рис. 4.9

(шрифт Arial, 32 пт, нормальный, выравнивание по левому краю). Первый слайд создан, однако он выглядит непривлекательно.

**1. Вставка картинок из коллекции клипов.** Вставим на него изображение компьютера из коллекции Microsoft Office. Для этого в меню **Вставка** выберем **Рисунок → Картинки**. В области задач появится меню **Коллекция клипов**.

Возможны два варианта поиска подходящей картинки. В поле **Искать** введем слово **компьютер** и нажмем кнопку **Начать**. Найденные картинки будут отображены в окне обзора (рис. 4.9, а). Если подходящая картинка не нашлась, следует просмотреть содержание всех коллекций, применив команду **Упорядочить картинки**. В результате появится окно, в левой части которого отображается список папок с коллекциями, а в правой — содержание выбранной коллекции (рис. 4.9, б).

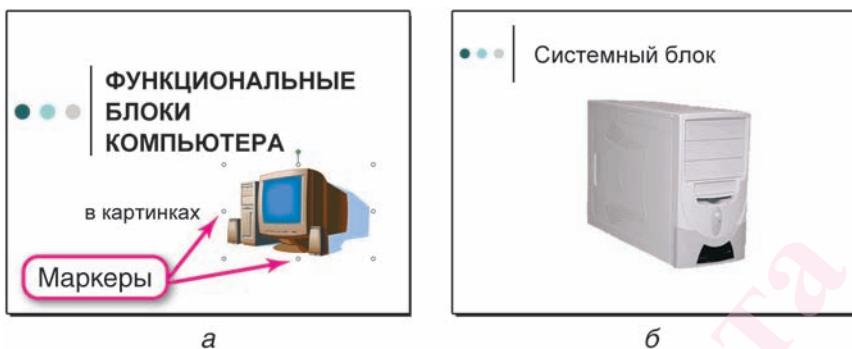
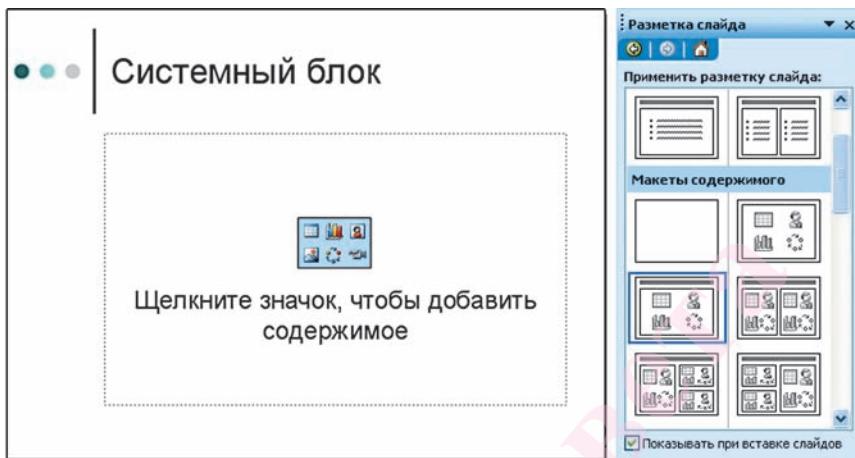


Рис. 4.10

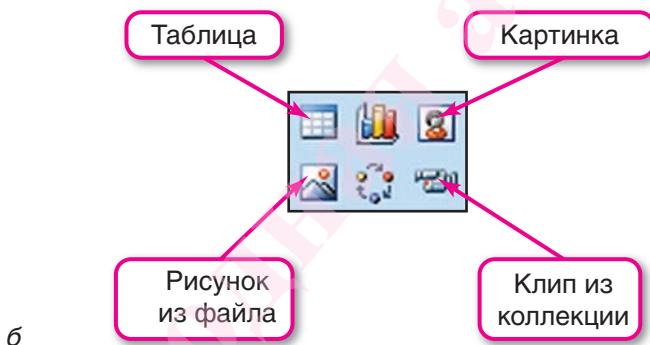
Выберем подходящую картинку и двойным щелчком мыши или перетаскиванием с нажатой левой кнопкой мыши разместим ее на слайде (рис. 4.10, а).

Расположение и размер картинки регулируются при помощи маркеров, которые появляются при выделении картинки щелчком мыши (см. рис. 4.10, а). Вид указателя мыши изменяется при подведении к маркеру. Стрелки  $\uparrow$  и  $\leftrightarrow$  указывают на возможность изменения размеров по вертикали и горизонтали, крестик  $\oplus$  — на перемещение, стрелка в виде дуги  $C$  — на возможность поворота. Установим положение и размеры картинки в соответствии с рисунком 4.10, а.

**2. Вставка рисунков из файла.** Создадим второй слайд, используя тот же шаблон оформления. Слайд должен содержать заголовок **Системный блок** и его фотографию. Для вставки изображения используем меню **Вставка** → **Рисунок** → **Из файла**. В окне **Добавление рисунка** выберем папку (например, **рисунки**) и файл (например, **sblock.jpg**) и нажмем **Вставить**. С помощью маркеров подберем положение и размеры картинки в соответствии с рисунком 4.10, б.



а



б

Рис. 4.11

Вставка объектов упрощается, если в области **Разметка слайда** выбрать подходящий макет. В нашем примере это макет **Заголовок и объект** (рис. 4.11, а). Выбрав его, на слайде мы видим миниатюру со значками (рис. 4.11, б). Чтобы выбрать нужный объект, достаточно щелкнуть мышью на его значке.

Таким способом добавим к презентации еще два слайда: **Монитор** (рис. 4.12, а) и **Клавиатура и мышь** (рис. 4.12, б).



Рис. 4.12

Сохраним и продемонстрируем созданную презентацию.

- ? 1. Что называется клипартом?
- 2. Как найти картинку в коллекции?
- 3. Как вставить картинку из коллекции?
- 4. Как вставить рисунок из файла?

### ***Упражнение***

Создайте презентацию с использованием рисунков из файлов и картинок из коллекций по одной из тем: «Моя Родина — Беларусь», «Минск — город-герой», «Мой родной город», «Природа родного края», «Животные», «Растения», «Птицы», «Галерея ученых (писателей, художников, артистов, музыкантов)».

## **§ 18. Использование декоративного текста**

Для художественного оформления текста на слайдах, например заголовков, используется **декоративный** (или **фигурный**) текст. Декоративный текст является графическим объектом WordArt. Его можно деформировать, наклонять, поворачивать, растягивать, добавлять тень, вписывать в фигуры.

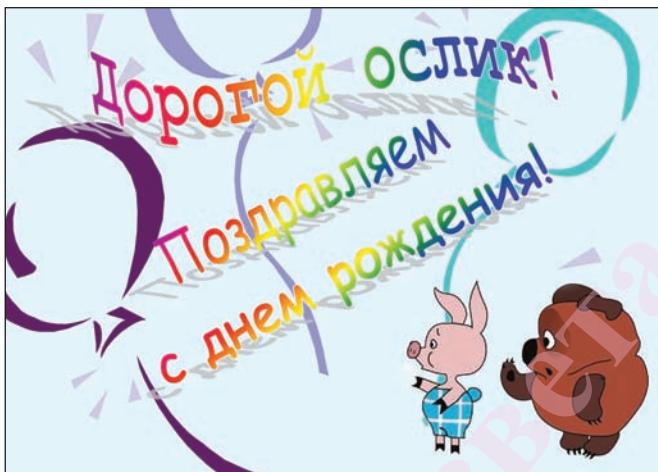


Рис. 4.13

Пример. Создать поздравительную открытку (рис. 4.13) с использованием декоративного текста.

Откроем редактор PowerPoint.

В области задач **Дизайн слайда** выберем подходящий шаблон оформления, например **Шары**.

Для вставки декоративного текста воспользуемся панелью инструментов **Рисование** (рис. 4.14). Нажмем кнопку **Добавить объект WordArt** .



Рис. 4.14

Из предложенных образцов (рис. 4.15) выберем стиль надписи. В появившемся диалоговом окне **Изменение текста WordArt** установим шрифт (на-



Рис. 4.15

пример, CourierNew, 36 пт, жирный) и вместо слов **Текст надписи** введем обращение **Дорогой ослик!** (рис. 4.16).

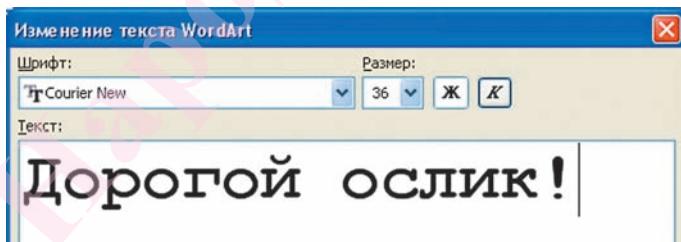


Рис. 4.16

Настраивать параметры декоративного текста удобно с помощью панели WordArt (рис. 4.17), которая вызывается командами **Вид → Панели инструментов → WordArt**.

Рис. 4.17



Размещение, размеры и форму декоративного текста будем изменять при помощи маркеров, которые появляются при выделении объекта (рис. 4.18).



Рис. 4.18

Аналогичным способом вставим декоративный текст **Поздравляем с днем рождения!**

В качестве подписи на нашей открытке вставим подготовленные заранее рисунки. Для этого воспользуемся панелью инструментов **Рисование** (или используем меню **Вставка → Рисунок → Из файла**). С помощью маркеров отрегулируем положение и размеры изображений в соответствии с рисунком 4.13.

Сохраним и продемонстрируем созданную открытку.

- ?
1. Для чего используется декоративный текст?
  2. Как вставить декоративный текст?
  3. Как изменить размещение и форму декоративного текста?

### *Упражнение*

Создайте открытку с использованием декоративного текста, рисунков из файлов и картинок из коллекций к одному из праздников: Новый год, 8 Марта, День Победы, День учителя.

## § 19. Использование фигур

На слайдах нередко используются геометрические фигуры, стрелки, указатели. В программах Microsoft Office такие объекты называются **автофигурами**. Они сгруппированы по категориям: **Линии**, **Основные фигуры**, **Фигурные стрелки**, **Звезды и ленты**, **Выноски** и др. (рис. 4.19).

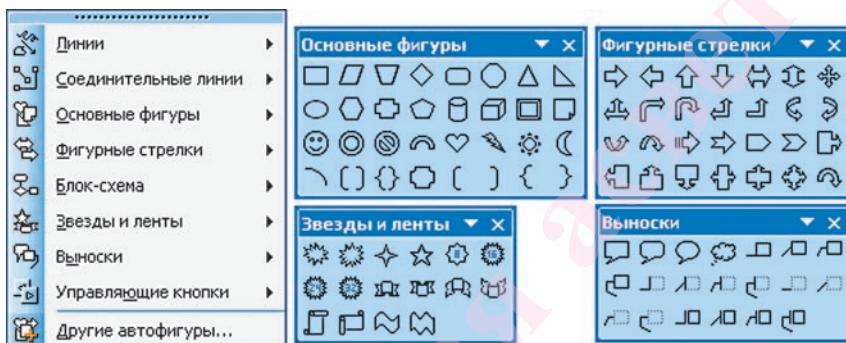


Рис. 4.19

Для выбора фигур используют меню **Вставка** → **Рисунок** → **Автофигуры** или панель инструментов **Рисование** (см. рис. 4.14). С помощью панели инструментов можно изменять цвет и способ заливки фигур, цвет и толщину линий, тип штриха, вид стрелки, стиль тени и даже делать фигуры объемными. При выделении фигуры появляются маркеры, позволяющие изменять геометрические свойства фигур, например размер и ориентацию.

Фигуры могут использоваться для улучшения структуры и выразительности представляемой информации, а могут иметь и самостоятельное значение, например в презентациях по геометрии, рисованию, черчению.

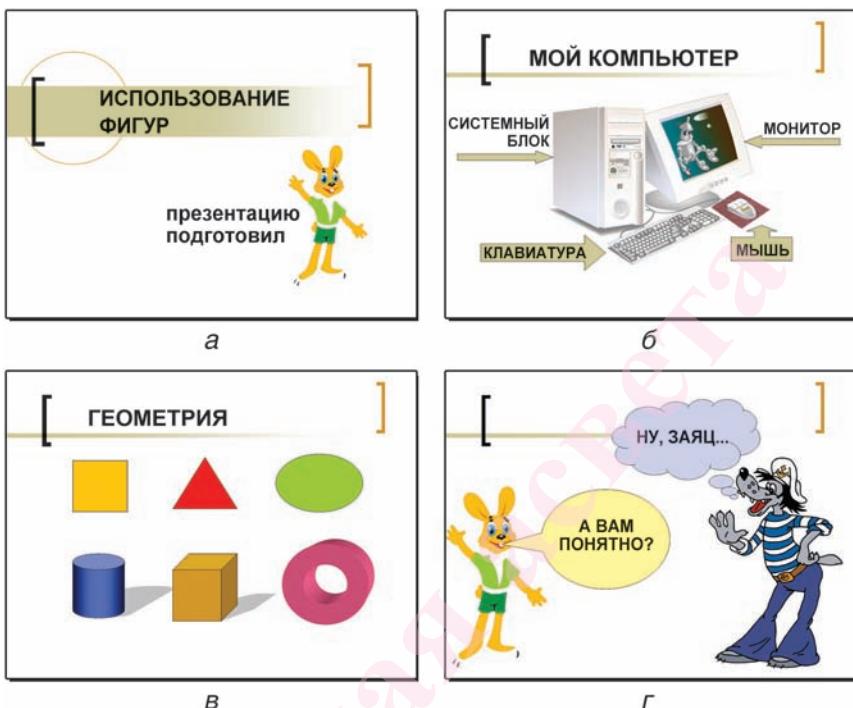


Рис. 4.20

Пример. Создать презентацию с использованием фигур.

Откроем редактор PowerPoint. Выберем подходящий шаблон оформления, например Идея. Оформим титульный слайд в соответствии с рисунком 4.20, а. Вставим рисунок зайца из файла.

Создадим второй слайд, используя макет оформления Заголовок и объект. Напишем заголовок Мой компьютер. Вставим фотографию из файла (рис. 4.20, б).

Для вставки стрелок-указателей воспользуемся панелью инструментов Рисование. Выберем подходящие стрелки и нарисуем их на слайде (см. рис. 4.20, б). Подберем цвет заливки и размеры фигур.

Названия **системный блок** и **монитор** разместим в надписях над стрелками, вставляя их с помощью кнопки **Надпись**  панели **Рисование**. Названия **клавиатура** и **мышь** напишем на стрелках. Для вставки текста в фигуру щелчком мыши выделим ее, правой кнопкой мыши вызовем контекстное меню и выберем пункт **Добавить текстовую строку**. Размер и начертание текста устанавливаются так же, как в текстовом редакторе.

На третьем слайде с заглавием **Геометрия** нарисуем следующие фигуры: желтый квадрат, красный треугольник, зеленый овал, синий цилиндр, коричневый куб и розовое кольцо (рис. 4.20, в).

На четвертый слайд поместим рисунки волка и зайца из файла. Для имитации диалога добавим фигуры **Выноски**: для зайца — **овальную**, а для волка — **облако** (рис. 4.20, г).

Сохраним и продемонстрируем созданную презентацию.

- ? 1. Какие объекты называются автофигурами?  
2. Как вставить фигуру на слайд?  
3. С какой целью можно использовать фигуры в презентации?

### Упражнение

Создайте презентацию с использованием фигур, рисунков из файлов и картинок из коллекций по одной из тем: «День и ночь» (исользуйте фигуры: Солнце, облака, месяц, звезды), «Диалог сказочных персонажей» (исользуйте выноски), «Геометрические фигуры».

## § 20. Элементы мультимедиа

Использование элементов мультимедиа делает презентацию наиболее яркой и наглядной. Под **мультимедиа** понимают компьютерную технологию, обеспечивающую с помощью технических и программных средств совместное использование текста, графики, звука, анимации и видео. Компьютерные презентации могут содержать звуковые и анимационные эффекты, речевое и музыкальное сопровождение, а также видеофрагменты.

В редакторе PowerPoint предусмотрено несколько вариантов звукового сопровождения презентаций. Записанные ранее звуковые фрагменты могут быть вставлены из файла или из коллекции (объекты клипарта). Можно скопировать музыку с компакт-диска или записать с помощью микрофона, если компьютер оснащен звуковой картой.

Пример 1. Создать поздравительную открытку с музыкальным сопровождением (рис. 4.21).



Рис. 4.21

Откроем редактор PowerPoint. Оформим слайд в соответствии с рисунком 4.21, вставив в качестве фона подходящее изображение из файла. Декоративным текстом напишем поздравление **С Новым годом!**

Используем музыкальный фрагмент, например один куплет песни «В лесу родилась елочка», сохраненный в файле **елочка.wav**. Для его вставки в презентацию воспользуемся меню **Вставка** → **Фильмы и звук** → **Звук из файла**, выберем файл и укажем способ воспроизведения **Автоматически** (рис. 4.22). В этом случае музыка начнет звучать сразу же после начала показа презентации. На наличие музыкального сопровождения указывает появившийся на слайде значок . Можно изменить его размеры или перетащить в другое место. Сохраним и продемонстрируем созданную презентацию.

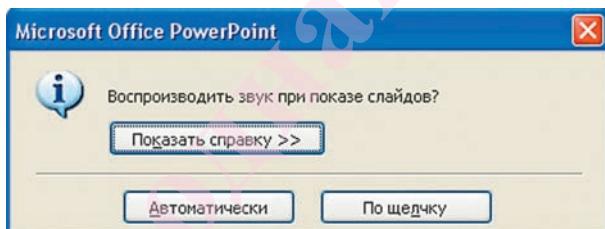


Рис. 4.22

**Пример 2.** Создать слайд с использованием звукозаписи.

Продублируем созданный в примере 1 слайд с помощью меню **Вставка** → **Дублировать слайд**. В результате оба слайда будут иметь одинаковое оформление и музыкальное сопровождение. Удалим музыкальный фрагмент из второго слайда. Для этого выделим значок и нажмем клавишу Del. С помощью меню **Вставка** → **Фильмы и звук** → **Записать звук**

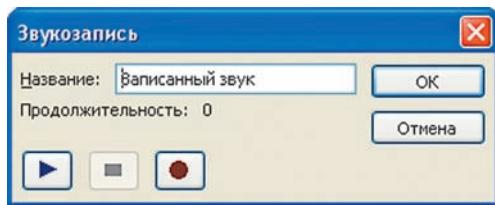


Рис. 4.23

вызовем панель **Звукозапись** (рис. 4.23). Нажатием красной кнопки включим режим записи и в подключенный заранее микрофон произнесем поздравление: «С Новым годом! С новым счастьем!» Нажатием черной кнопки остановим запись (ее продолжительность около 5 с). Сохраним и продемонстрируем созданную презентацию.

**Пример 3.** Создать фрагмент рекламы мобильного телефона с демонстрацией звукового сигнала по желанию пользователя.

Создадим слайд в соответствии с рисунком 4.24, вставив изображение мобильного телефона из файла и декоративный текст.



Рис. 4.24

Для вставки звука телефона из коллекции MS Office воспользуемся меню **Вставка** → → **Фильмы и звук** → **Звук из коллекции картинок**. Из списка файлов мультимедиа выберем **Телефон.wav** (см. рис. 4.24). Укажем способ воспроизведения **По щелчку** (см. рис. 4.22). В этом случае во время демонстрации слайда звук будет воспроизводиться только после щелчка мышью на значке . Продемонстрируем презентацию.

Изменим параметры воспроизведения звука. Для этого щелчком правой кнопкой мыши на значке вызовем контекстное меню и выберем пункт **Изменить звуковой эффект**. В окне **Параметры звука** выполним установки, показанные на рисунке 4.25. Теперь в ходе показа презентации звук будет воспроизводиться автоматически, а его значок станет невидимым. Продемонстрируем презентацию.

С помощью аналогичных действий в презентацию вставляются видеофрагменты.

- ?** 1. Что понимают под мультимедиа?
- 2. Как вставить звук из файла? Из коллекции клипов?
- 3. Как записать звук?

### *Упражнение*

Создайте презентацию с использованием звука, рисунков из файлов и картинок из коллекций по одной из тем: «Звуковая открытка», «Любимые мелодии (музыканты, певцы, группы)», «Голоса природы», «Певчие птицы».

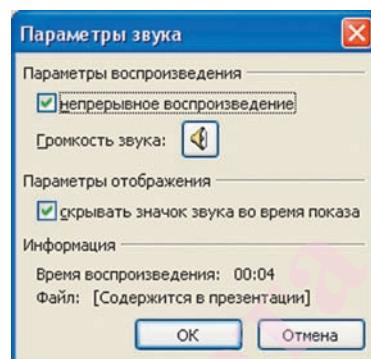


Рис. 4.25

## §21. Использование эффектов

Редактор PowerPoint предоставляет разработчику большой набор разнообразных визуальных и звуковых эффектов. Эффекты используются при переходе от слайда к слайду или при перемещении отдельных объектов слайда в процессе показа презентации. Например, можно создать эффекты появления и движения надписей и фигур или добавить звук аплодисментов при появлении рисунка. Визуальные и звуковые эффекты призваны помочь изложению материала презентации, подчеркивая отдельные моменты. В то же время чрезмерное использование эффектов мешает восприятию основного содержания презентации. Поэтому злоупотреблять ими не следует.

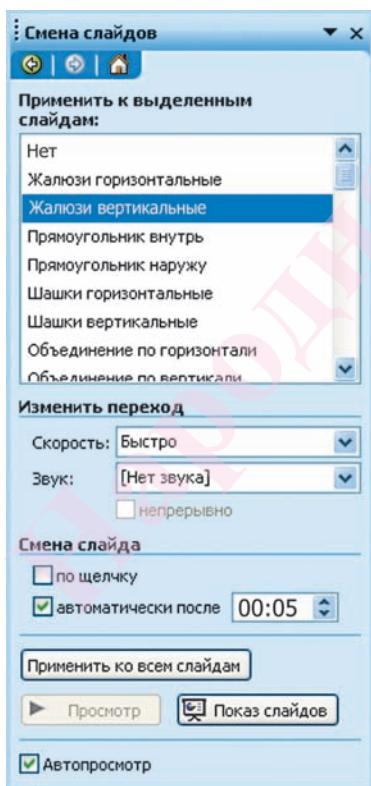


Рис. 4.26

**Пример 1.** Настроить смену слайдов презентации «Функциональные блоки компьютера» (см. рис. 4.10, 4.12).

Откроем презентацию. Выделим первый слайд. Применим к нему эффект **Прямоугольник наружу**, способ смены слайда **по щелчку**. Зададим способ смены второго и третьего слайдов **автоматически после 5 с**, эффект **Жалюзи вертикальные** (см. рис. 4.26). Продемонстрируем презентацию.

Настройка анимации отдельных объектов на слайде выполняется с помощью меню **Показ слайдов** → **Настройка анимации**. Последовательно выделяются объекты слайда, и в окне **Настройка анимации** задается порядок и характер их появления и движения. Командой **Добавить эффект** настраивают анимационный и звуковой эффекты, скорость движения и другие параметры (рис. 4.27).

**Пример 2.** Настроить эффекты анимации объектов презентации «Мобильные телефоны» (см. рис. 4.24).

Откроем презентацию. Выделим фигурный текст (объект WordArt). В окне **Настройка анимации** выберем **Добавить эффект** → **Вход** → **Жалюзи**. Уста-

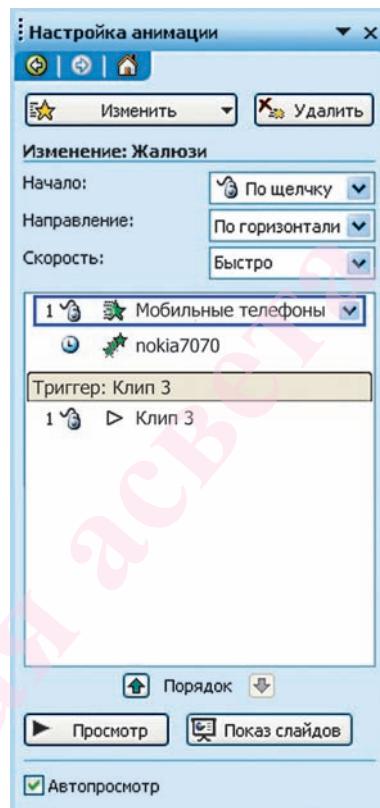


Рис. 4.27

новим параметры: **Начало** — По щелчку, **Направление** — По горизонтали, **Скорость** — Быстро (см. рис. 4.27). Проверим созданный эффект нажатием кнопки **Просмотр**.

Настройка эффектов анимации для рисунков и фигур выполняется аналогично. Выделим изображение мобильного телефона. В окне **Настройка анимации** выберем **Добавить эффект** → **Вход** → **Вылет**. Установим параметры: **Начало** — После предыдущего, **Направление** — Снизу, **Скорость** — Средне.

Продемонстрируем презентацию. Все элементы слайда должны показывать заданное настройками поведение.

**Пример 3.** Создать слайд «Я строю дом» с использованием фигур и эффектов анимации.

Откроем редактор PowerPoint. Выберем подходящий шаблон оформления, например **Сетка с тенью**.

Разместим на слайде рисунки зайца и волка из файлов. Для диалога используем фигуры из категории **Выноски с текстом** Я строю дом и Ну, заяц... Нарисуем и разместим в разных местах слайда «строительные блоки»: крышу, стену и окно (фигуры треугольник и прямоугольники) (рис. 4.28, а).



Рис. 4.28

Зададим эффекты. Пусть дом собирается перемещением стены и крыши, а окно не двигается и служит ориентиром для установки стены (рис. 4.28, б).

Выделим прямоугольник «стена». В области задач **Настройка анимации** выберем **Добавить эффект** → **Пути перемещения** → **Пользовательский путь** → **Линия**. Нарисуем отрезок от середины стены до середины прямоугольника «окно» (рис. 4.29). Проверим эффект нажатием кнопки **Просмотр**.

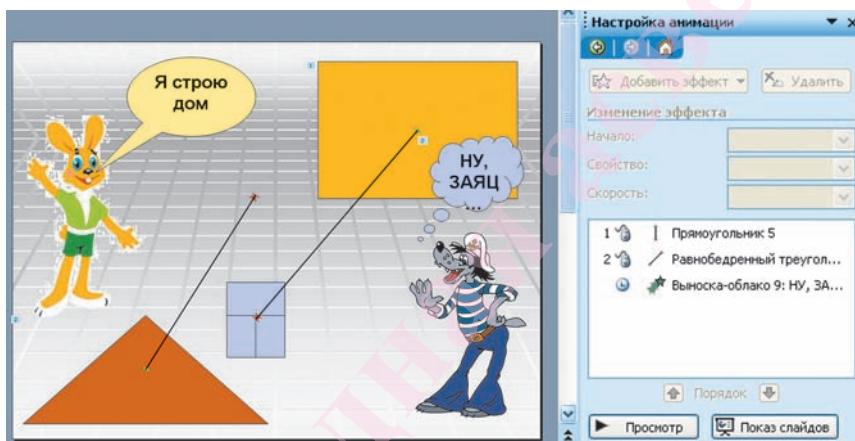


Рис. 4.29

Точно указать линией конечное положение крыши трудно. Выделим треугольник «крыша» и выполним следующие действия: **Добавить эффект** → **Пути перемещения** → **Вправо вверх**. Щелчком мыши выделим появившуюся стрелку. Ее начало обозначено зеленым, а конец красным маркерами. Подберем положение красного маркера так, чтобы крыша устанавливалась точно на стену.

Выделим фигуру **Выноска** с текстом **Ну, заяц...**. Выберем эффект **Вход** → **Вылет**. Установим **Начало** — **После предыдущего**, **Направление** — **Сверху**.

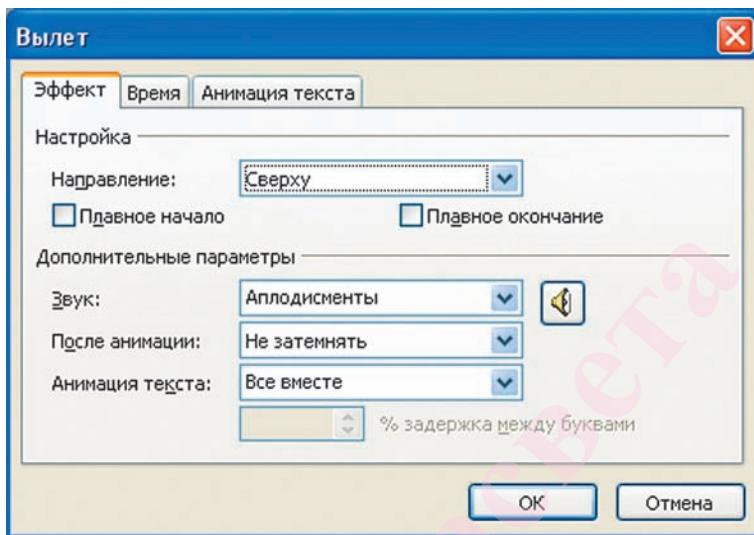


Рис. 4.30

Для задания звукового сопровождения настроим параметры эффекта. Щелчком мыши на кнопке рядом с именем **Выноска-облако 9...** вызовем диалоговое окно и выберем звук **Аплодисменты** (рис. 4.30).

Сохраним и продемонстрируем созданную презентацию.

- ?** 1. С какой целью в презентации используются эффекты?
- 2. Какие визуальные и звуковые эффекты возможны в презентациях?

### Упражнение

Создайте анимированную презентацию с использованием текста, фигур, рисунков из файлов и картинок из коллекций по одной из тем: «Смена дня и ночи» (два слайда, фигуры Солнце, облака, месяц, звезды), «Времена года», «Анимация фрагмента сказки» (диалог сказочных персонажей).

## § 22. Создание презентаций по темам учебных предметов

Компьютерные презентации играют важную роль в образовании. Рассмотрим примеры слайдов и презентаций по темам учебных предметов.

Пример 1. Создать презентацию «Животные родного края».

Откроем редактор PowerPoint. Выберем подходящий шаблон оформления, например Глобус.

Оформим титульный слайд в соответствии с рисунком 4.31, а: поместим на него название презентации, изображение карты Республики Беларусь и уменьшенные фотографии животных, например аиста, зубра, зайца, белки, совы. На следующие слайды будем помещать названия животных и их крупные фотографии (рис. 4.31, б).



Рис. 4.31

Пример 2. Создать презентацию «Бабочка на цветке» с видеофрагментом.

Создадим слайд в соответствии с рисунком 4.32. Для вставки видеофрагмента воспользуемся меню Вставка → Фильмы и звук → Фильм из файла (или Фильм из коллекции клипов), выберем файл и ука-

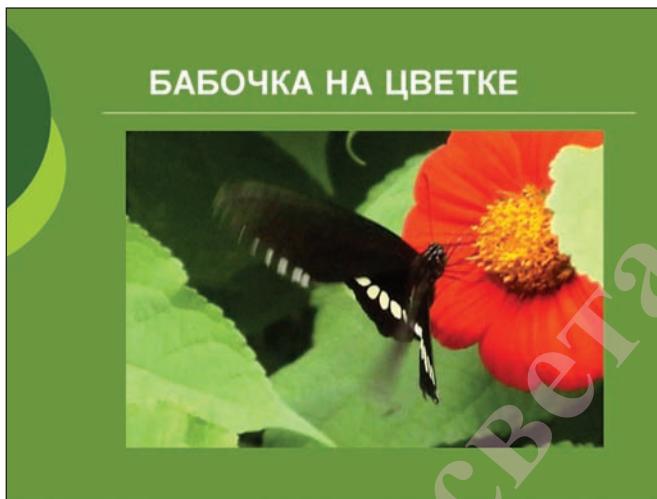


Рис. 4.32

жем способ воспроизведения фильма, например **По щелчку** (рис. 4.33). На слайде будет отображаться первый кадр фильма (см. рис. 4.32). Можно изменить его размеры и расположение. Во время демонстрации слайда фильм будет воспроизводиться после щелчка мышью по области кадра. Сохраним и продемонстрируем созданную презентацию.

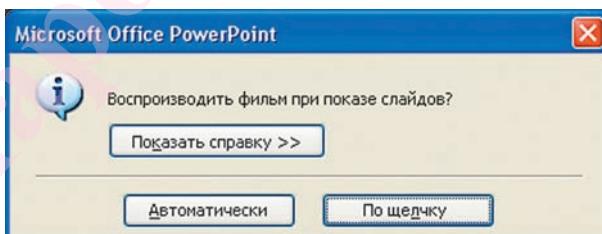


Рис. 4.33

**Пример 3.** Создать презентацию «Построение точки по ее координатам».

Откроем редактор PowerPoint. Возьмем макет слайда **Только заголовок**.

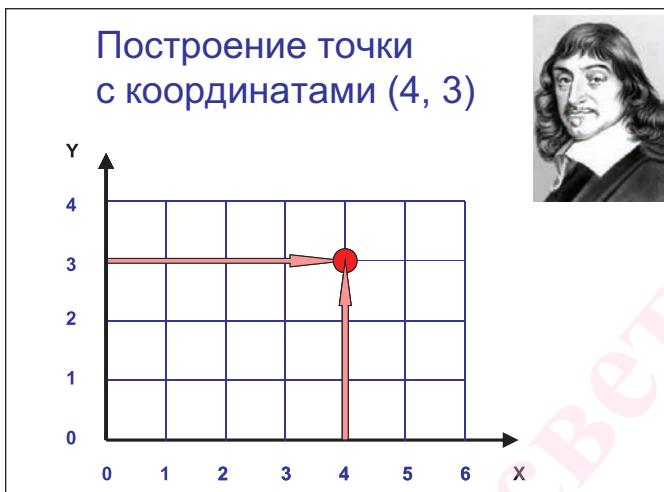


Рис. 4.34

Введем заголовок **Построение точки с координатами (4, 3)**. Вставим рисунок из файла, например портрет математика Рене Декарта.

С помощью фигур **Стрелка** и **Линия** нарисуем оси координат и сетку. Вставим надписи с делениями (рис. 4.34). Для размещения линий на равных расстояниях во время рисования покажем вспомогательную сетку нажатием кнопки на панели инструментов **Стандартная**.

Для демонстрации построения точки с заданными координатами используем фигуры, настроив эффекты: **Стрелка вверх** — Вылет снизу по щелчку (абсцисса), **Стрелка вправо** — Вылет слева после предыдущего (ордината), **Овал** — Вылет сверху после предыдущего (точка).

Сохраним и продемонстрируем созданную презентацию.

**Пример 4.** Создать презентацию «Движение Луны вокруг Земли».



Рис. 4.35

Откроем редактор PowerPoint. Оформим слайд в соответствии с рисунком 4.35. Вставим изображения звездного неба, Земли и Луны. Выделим Луну и настроим эффекты анимации: Траектория — Круг, Скорость — Средне, Повторение 4 раза.

Сохраним и продемонстрируем созданную презентацию.

### *Упражнение*

Создайте презентацию по одной из тем:

**История:** «Средневековые замки», «Культура Возрождения», «Грюнвальдская битва», «Изобретение книгопечатания», «Франциск Скорина».

**География:** «Великие географические открытия», «Рельеф Земли», «Кругооборот воды», «Погода и климат», «Население Земли».

**Биология:** «Строение клетки», «Грибы», «Многообразие растений», «Строение цветка», «Многообразие животных», «Растения», «Птицы», «Природа родного края».

**Математика:** «Определение координат точки на плоскости», «График линейной функции», «Геометрические фигуры», «Медиана, биссектриса и высота треугольника», «Свойства и признаки равнобедренного треугольника».

**Физика:** «Механическое движение», «Силы в природе», «Работа и энергия», «Простые механизмы», «Трение в природе и технике».

**Химия:** «Химия вокруг нас», «Атомы и молекулы», «Превращения веществ», «Кислород в природе», «Вода в жизни человека».

## Глава 5

# КОМПЬЮТЕРНЫЕ КОММУНИКАЦИИ

## И СЕТЬ ИНТЕРНЕТ

### **§ 23. Глобальная компьютерная сеть Интернет**

Интернет представляет собой всемирную компьютерную сеть или сообщество сетей, которые связывают между собой в единое целое миллионы различных устройств из разных уголков мира. К сети Интернет могут подключаться: персональные и мобильные компьютеры, телефоны, устройства, которые установлены в телевизорах, автомобилях, музыкальных центрах и игровых приставках. С каждым днем все больше устройств могут быть подключены к сети Интернет. Их доступ к сети обеспечивают организации-поставщики услуг Интернет — **интернет-провайдеры**.

Сеть Интернет предоставляет пользователю огромное количество полезной научной и технической информации, а также образовательные ресурсы, например электронные фонды и каталоги крупнейших библиотек и музеев мира, научные базы данных и поисковые системы, сведения о достижениях крупных научных и образовательных центров, научные статьи, рефераты. В сети Интернет отражаются политические события, новости бизнеса, медицины, культуры, спорта, туризма. Сегодня с помощью сети Интернет можно познакомиться с ассортиментом товаров в магазинах и даже купить их, узнать прогноз погоды, курсы валют, программы теле- и радиопередач, прочитать электронные газеты и журналы, скачать игры и компьютерные программы, видеофильмы и музыкальные произведения. Сеть Интернет обес-

печивает возможность общения людей в письменной и устной форме, проведения онлайн-конференций и дискуссий, поиска друзей и единомышленников.

Однако в сети Интернет можно встретить множество бесполезной или поверхностной информации, например навязчивую рекламу, литературные произведения низкого качества, сведения, несоответствующие действительным фактам.

Основные возможности сети Интернет реализуются через ее службы (или сервисы), среди которых наиболее популярны *Всемирная информационная паутина, электронная почта, средства обмена сообщениями в режиме реального времени* и др.

**Всемирная информационная паутина (World Wide Web — WWW)** — на сегодняшний день самая популярная и удобная служба сети Интернет. Она объединяет большинство услуг этой сети. Информация Всемирной паутины организована в виде гипертекстовых документов (веб-страниц), содержащих не только тексты, но и графику, анимации, аудио- и видеофрагменты.

Одной из наиболее распространенных и эффективных служб сети Интернет является **электронная почта (Electronic Mail — E-mail)**. Ее назначение — обмен текстовыми сообщениями (электронными письмами) с помощью электронных почтовых ящиков. С электронным письмом могут пересылаться прикрепленные к нему файлы, например с изображениями. В настоящее время эта услуга без дополнительной оплаты автоматически предоставляется всем, кто подключился к сети Интернет. Адрес почтового

ящика состоит из двух частей: имени и адреса провайдера, которые разделяются символом @, например cat@tut.by.

**Средства обмена сообщениями в режиме реального времени (Internet Relay Chat — IRC)** обеспечивают ведение письменного разговора (диалога) между пользователями сети в реальном времени. Все большую популярность приобретают аудиовизуальные средства общения, с помощью которых беседующие слышат и даже видят друг друга.

Подключение к компьютерной сети Интернет может осуществляться различными способами. Наиболее распространенными являются:

- доступ по телефонной линии с помощью модема;
- доступ по выделенному каналу связи;
- беспроводное (мобильное) подключение.

Доступ к информационным ресурсам сети Интернет и их просмотр (навигация) осуществляется с помощью программ, которые называются **браузерами** (от англ. *browse* — пролистать, проглядеть) или **веб-обозревателями**. Наиболее популярные браузеры: Microsoft Internet Explorer, Opera, Mozilla Firefox, Netscape Communicator.

Создание Всемирной информационной паутины (WWW) было вызвано необходимостью объединения различных услуг сети Интернет и расширения возможностей доступа к различным информационным ресурсам сети.

Документы, с которыми работает служба WWW, содержат гипертекстовую информацию. Под **гипертекстом** понимается способ организации информации, который использует **гиперссылки**. Гиперссылки могут указывать на информацию внутри до-

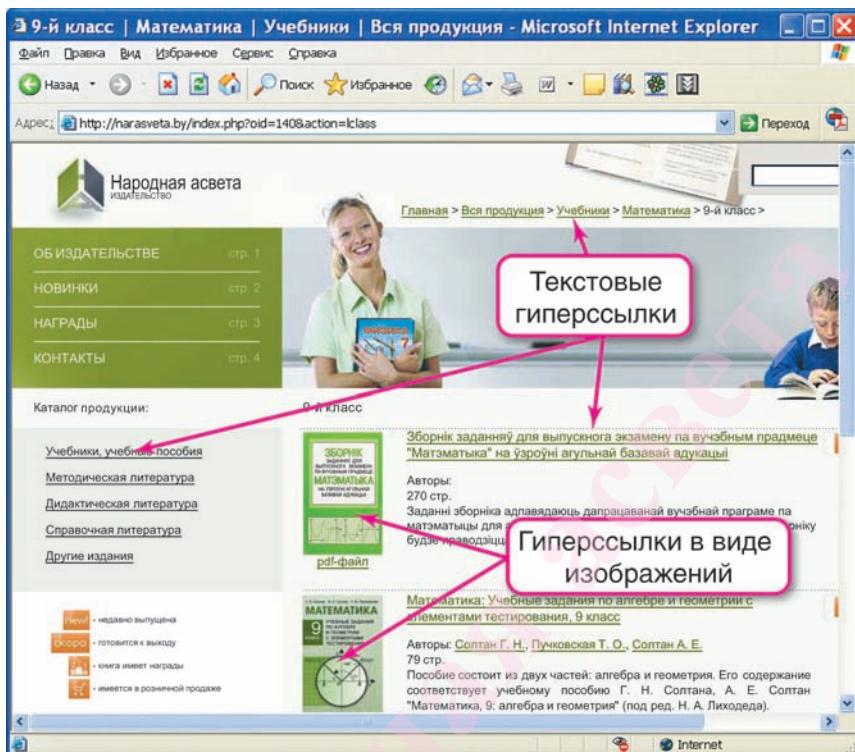


Рис. 5.1

кумента, а могут связывать данный документ с различными объектами сети Интернет: файлами, другими гипертекстовыми документами, содержащими текст, изображения, анимацию, видео- и аудиофрагменты.

Гиперссылки могут быть оформлены в виде текста или изображения (рис. 5.1).

При создании гипертекстовых документов используют специальный язык **HTML** (HyperText Markup Language — язык разметки гипертекста). Обычно HTML-документы имеют расширение **.htm** или **.html**.

Размещенный во всемирной паутине гипертекстовый документ называют **веб-страницей** (web-страницей). Каждая веб-страница имеет свой адрес, зная который, ее можно просмотреть с помощью браузера.

Набор связанных гиперссылками веб-страниц, объединенных общими целями и содержанием, называют **веб-сайтом** или просто **сайтом**.

Правила передачи веб-страниц по сети и их просмотра на экране компьютера определены в протоколе передачи гипертекста **HTTP** (HyperText Transfer Protocol). С помощью протокола HTTP могут передаваться практически любые данные: графика, музыка, видео.

Чтобы посмотреть, куда указывает гиперссылка, во многих случаях достаточно навести на нее указатель мыши и в строке состояния прочитать адрес. Более подробные сведения можно получить, если щелкнуть на ссылке правой кнопкой мыши и из появившегося контекстного меню выбрать пункт **Свойства**. В результате откроется окно, содержащее требуемую информацию.

Страницы веб-сайта хранятся на компьютере, который имеет специальное программное обеспечение и называется **веб-сервером**.

Рассматривая сайт, как совокупность веб-страниц, заметим, что он имеет главную страницу, которая содержит ссылки на другие веб-страницы сайта. Для организации доступа к ним достаточно знать адрес главной страницы сайта. Поэтому **адресом сайта** считают адрес его главной страницы на веб-сервере.

Запись адреса сайта, как правило, начинается с указания службы **www**. Затем следует несколько групп символов (слов), разделенных точками. Заканчивается адрес указанием на государство (**by** — Беларусь, **ru** — Россия, **ua** — Украина, **pl** — Польша, **fr** — Франция) или принадлежность (**org** — организация, **edu** — образование, **com** — коммерческий сайт). Примеры адресов сайтов:

**www.adu.by**,

**www.tut.by**,

**www.bspu.unibel.by**.

Адрес веб-сайта может предваряться именем протокола с обязательными знаками **://** (**http://**). Адрес может содержать путь к конкретному документу (файлу или папке) с указанием имени, например **http://7wonders.worldstreasure.com/index.htm**.

- ?** 1. Какие службы используются для реализации основных возможностей сети Интернет?
2. Какие возможности предоставляет пользователю сети Интернет служба WWW?
3. Каково назначение электронной почты?
4. Что называется веб-страницей?
5. Что называют веб-сайтом?

## § 24. Просмотр веб-страниц

### 24.1. Элементы интерфейса браузера

**Браузер** — это программа, предназначенная для навигации в сети Интернет. Во время своей работы браузер по заданному адресу обращается к веб-серверу, запрашивает веб-страницу, получает информацию, преобразует и отображает ее на экране компьютера.



Рис. 5.2



Рис. 5.3

Рассмотрим работу с браузером Microsoft Internet Explorer, который входит в состав операционной системы Windows.

Его можно запустить щелчком мыши на значке (рис. 5.2) на Панели задач, двойным щелчком мыши на ярлыке (рис. 5.3) на Рабочем столе, или через систему меню кнопки Пуск.

Интерфейс программы достаточно прост и направлен на то, чтобы пользователь смог быстро освоить процесс открытия и просмотра веб-страниц. После запуска браузера Internet Explorer на экране компьютера открывается окно, основные элементы которого показаны на рисунке 5.4.

**В Строке заголовка** размещается название просматриваемого документа. **Адресная строка** содержит его адрес.

Основной частью окна обозревателя является **Область просмотра документа**. В ней размещается просматриваемая веб-страница.

**Строка меню** обеспечивает доступ к необходимым для работы с браузером командам. Некоторые из разделов меню, например **Файл** и **Правка**, являются общими по своему функциональному назначению для всех офисных приложений.

Быстрый доступ к наиболее часто используемым командам браузера обеспечивает **Панель инструментов**.

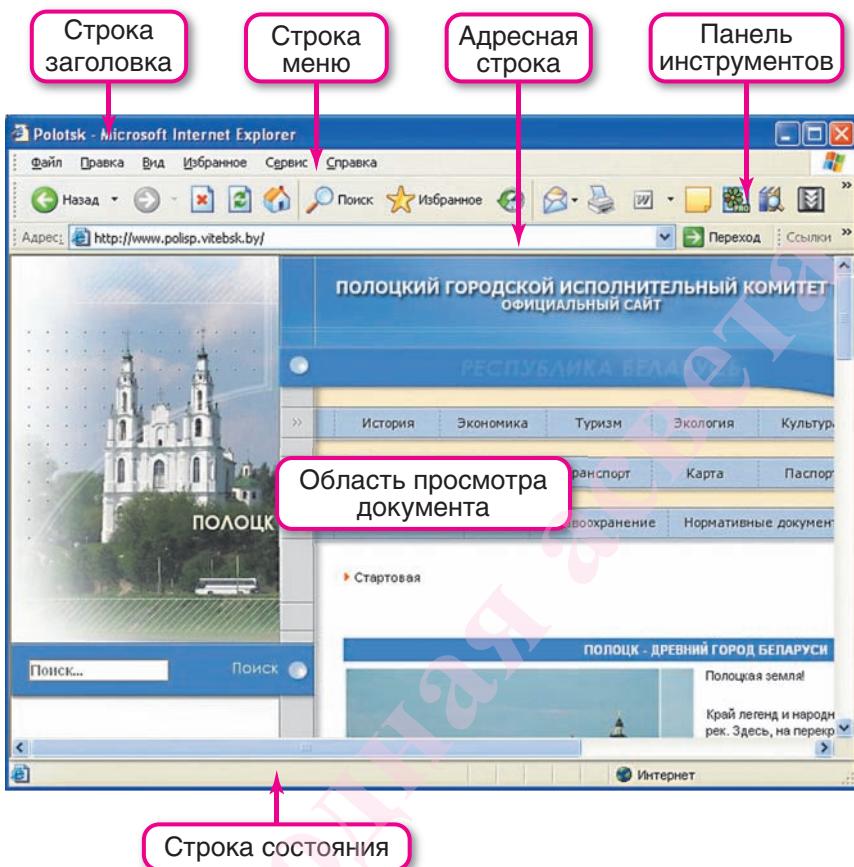


Рис. 5.4

тов. В Строке состояния отражается ход загрузки документа и другая информация.

После запуска браузера в Области просмотра документа, как правило, отображается **домашняя страница**. В качестве домашней используют пустую страницу или главную страницу наиболее посещаемого сайта. Установка домашней страницы выполняется с помощью меню **Сервис → Свойства обозревателя**. Для этого на вкладке **Общие**

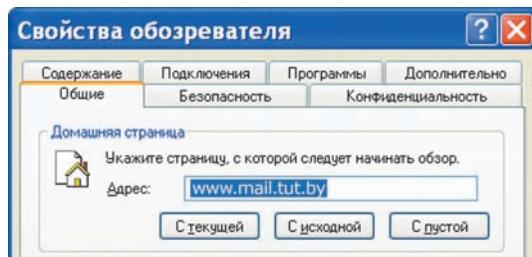


Рис. 5.5

окна **Свойства обозревателя** указывают адрес страницы, которую хотят видеть в качестве домашней (рис. 5.5).

Совершая путешествия по сети Интернет, в любой момент можно вернуться на свою домашнюю страницу, щелкнув левой кнопкой мыши на значке **Домой**  на Панели инструментов (см. рис. 5.4).

## 24.2. Работа с браузером

Для открытия веб-страницы необходимо в адресной строке ввести адрес этой страницы или сайта, например, <http://7wonders.worldstreasure.com> (данний адрес указывает на веб-сайт **Семь чудес света**) и нажать клавишу Enter. Откроется главная страница сайта (рис. 5.6).

Для перехода на другую страницу щелкнем левой кнопкой мыши на текстовой ссылке **Висячие сады Вавилона**. В адресной строке браузера автоматически появится новый адрес: <http://7wonders.worldstreasure.com/sady/index.htm> и откроется требуемая страница (рис. 5.7). Эта страница содержит гиперссылки в виде кнопок и изображений.

Близкие по содержанию и оформлению ссылки нередко группируются в виде меню (см. рис. 5.7).

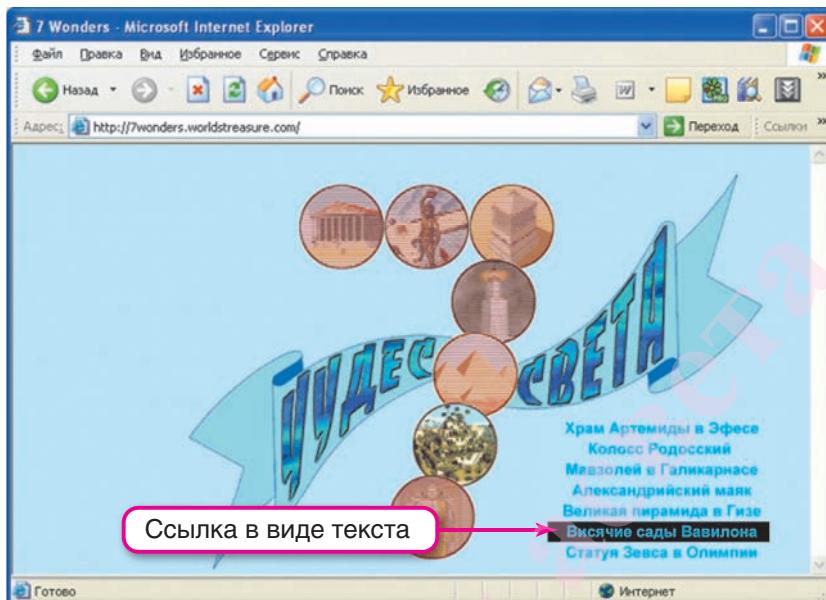


Рис. 5.6

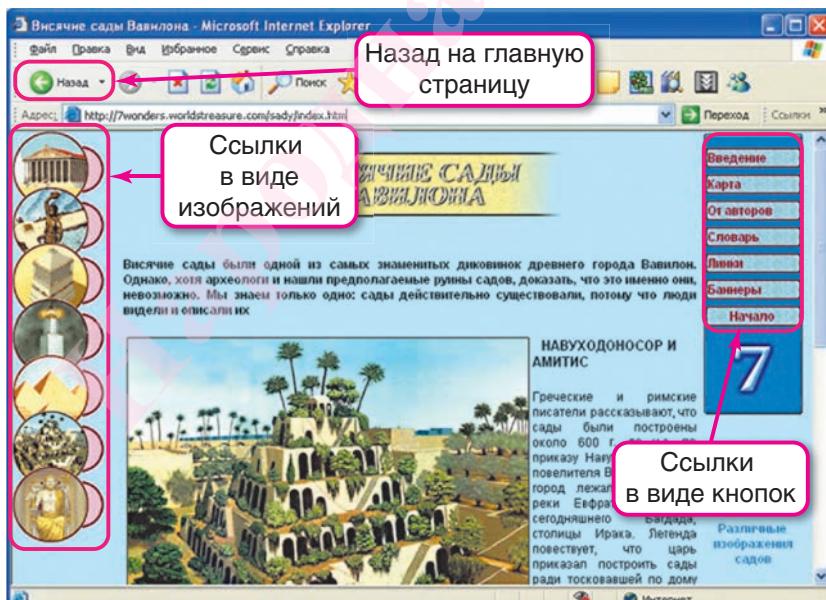


Рис. 5.7

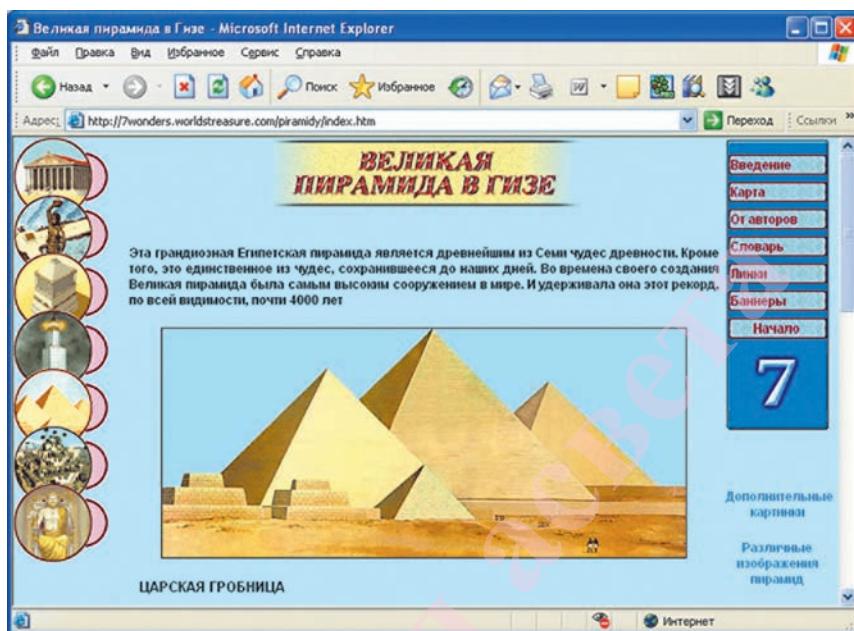


Рис. 5.8

Щелкнув левой кнопкой мыши на гиперссылке в виде изображения, перейдем на следующую страницу **Великая пирамида в Гизе** (рис. 5.8).

После загрузки страницы **Великая пирамида в Гизе** можно вернуться к предыдущей странице **Висячие сады Вавилона**, щелкнув мышью на стрелке влево (**Назад**) Назад . С этой страницы подобным способом можно вернуться на главную страницу сайта, или, щелкнув на стрелке вправо (**Вперед**) , — на следующую страницу из списка уже просмотренных.

При необходимости загрузку страницы можно остановить щелчком мыши на значке **Остановить** , повторно открыть страницу или обновить ее содержание можно щелчком мыши на значке **Обновить** .

Отправить открытый документ на печать можно щелчком мыши на значке **Печать**  или с помощью меню **Файл → Печать**.

Сохранить веб-страницу на диске можно с помощью меню **Файл → Сохранить как**. При этом предлагаются выбрать формат сохраняемого документа:

- **Веб-страница, полностью** (документ сохраняется в формате **.html**, при этом рисунки и другие объекты сохраняются в папке с тем же именем и расширением **.files**);
- **Веб-архив, один файл** (документ со всеми объектами упаковывается в один архивный файл);
- **Веб-страница, только HTML** (текст документа сохраняется в формате **.html**, а рисунки и другие объекты теряются);
- **Текстовый файл** (текст сохраняется в формате **.txt**, который можно прочитать с помощью текстового редактора Блокнот).

Документы, сохраняемые первыми тремя способами, в последующем можно просматривать в браузере без подключения к сети Интернет в автономном режиме.

Ссылки на регулярно посещаемые сайты (например, новости, прогноз погоды, репертуар кинотеатров, программы теле- и радиопередач) удобно помещать в папку **Избранное**. Для этого необходимо:

- Находясь на нужной веб-странице, щелчком правой кнопки мыши вызвать контекстное меню и выбрать пункт **Добавить в Избранное**.
- Присвоить гиперссылке на эту страницу имя, например **Погода на meteo.by**.

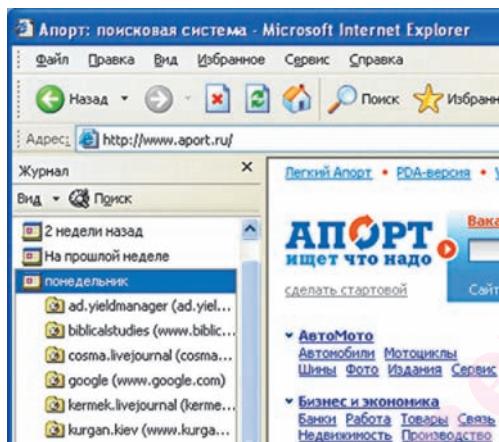


Рис. 5.9

Так можно коллекционировать адреса сайтов, создавая папки ссылок (например, **Виды Минска**). В эти папки можно помещать ссылки на постоянно обновляющиеся сайты. Это избавляет от повторного набора адресов и повышает удобство и оперативность работы.

Удаление ссылки из папки **Избранное** выполняется ее выделением и последующим нажатием клавиши Del.

Для быстрого открытия нужных страниц можно также использовать **Журнал**. В нем фиксируются адреса сайтов и страниц, посещенных за определенный период времени (рис. 5.9). Просмотреть **Журнал** можно с помощью значка

### 24.3. Вопросы безопасности в сети Интернет

Путешествуя по сети Интернет, Вы не задумываетесь, что это не всегда безопасно.

С ростом популярности сети Интернет начали проявляться негативные моменты, которые связаны

с бесконтрольным распространением информации. Появились сайты, пропагандирующие насилие и жестокость, расовую и национальную нетерпимость. Активно используют Интернет неофашистские организации и секты. Участились случаи финансовых афер и вымогательства.

Следует помнить, что с помощью специальных программ можно определить тип Вашего компьютера, операционную систему и даже адрес электронной почты.

Не раскрывайте информацию о себе, чтобы не стать жертвой атак назойливой рекламы, нежелательных сообщений (спама) или компьютерных мошенников, которые пытаются завладеть Вашиими личными данными (паролями, номерами кредитных карт). Некоторые вредоносные программы также используют интернет-соединение, чтобы завладеть Вашиими данными.

Для защиты информации следует соблюдать правила безопасности при работе в сети Интернет. Перечислим некоторые из них.

- Берегите пароли. Некоторые программы предлагают сохранить пароль, чтобы не вводить его каждый раз. Не делайте этого, особенно в программах, устанавливающих интернет-соединение.
- Не передавайте электронные адреса или пароли по электронной почте в открытом виде.
- Веб-страницы часто содержат диалоговые формы. Имейте в виду, что отправка данных веб-серверу небезопасна. Внимательно читайте тексты, требующие ответа или подтверждения. Не торопитесь нажать на кнопку «Да».

- Самый большой риск — это загрузка на Ваш компьютер файлов неизвестного происхождения. В сети Интернет можно найти большое количество бесплатных программ, игр, видеофильмов. Будьте осторожны, скачивая их, особенно с анонимных серверов. Ни в коем случае не запускайте их без предварительной проверки на вирусы.
- Не открывайте файлы, которые прикреплены к электронным сообщениям от незнакомых адресатов. Файлы и электронные письма могут содержать вредоносные программы.
- При работе в сети Интернет через локальную сеть не устанавливайте в качестве общего ресурса важные диски и папки (например, папку Windows).

Наиболее негативным проявлением воздействия сети Интернет на психику человека является интернет-зависимость. Некоторые люди настолько увлекаются виртуальным миром, что начинают предпочитать его реальному.

Не подвергайте себя неоправданному риску, посещая сомнительные сайты.

- ? 1. Для чего предназначен браузер?  
2. Назовите основные элементы окна браузера. Каково их назначение?  
3. Для чего используется домашняя страница и как ее установить?  
4. Как поместить ссылку в папку **Избранное**?  
5. Для чего в браузере используется **Журнал**?  
6. Какие правила безопасной работы в сети Интернет Вы знаете?

### *Упражнение*

Запустите браузер Internet Explorer. Откройте главные страницы веб-сайтов по указанию учителя.

Газеты и журналы	
Газета «Знамя юности»	<a href="http://www.zn.by">http://www.zn.by</a>
Журнал «Юный техник»	<a href="http://jtdigest.narod.ru">http://jtdigest.narod.ru</a>
Журнал «Юный натуралист»	<a href="http://www.unnaturalist.ru">http://www.unnaturalist.ru</a>
Библиотеки	
Национальная библиотека Беларуси	<a href="http://www.nlb.by">http://www.nlb.by</a>
Каталог электронных библиотек	<a href="http://allbest.ru/libraries.htm">http://allbest.ru/libraries.htm</a>
Образовательные ресурсы	
Занимательная математика — школьникам	<a href="http://www.math-on-line.com">http://www.math-on-line.com</a>
Увлекательный сайт о многогранном мире химии	<a href="http://www.alhimik.ru">http://www.alhimik.ru</a>
Олимпиадная информатика	<a href="http://olympiads.ru">http://olympiads.ru</a>
Музыка и культура	
Эрмитаж	<a href="http://www.hermitagemuseum.org">http://www.hermitagemuseum.org</a>
Музыкальный каталог	<a href="http://www.pesni.ru">http://www.pesni.ru</a>
Путешествия и туризм	
Интернет-каталог туристических услуг	<a href="http://www.tours.belti.ru">http://www.tours.belti.ru</a>
Туризм в Беларуси	<a href="http://belarustourism.by">http://belarustourism.by</a>

## Приложение

Таблица 1

### Зарезервированные (ключевые) слова Pascal ABC

Зарезервированное слово	Перевод на русский язык	Зарезервированное слово	Перевод на русский язык
And	И	Integer	Целый
Array	Массив	Line	Линия
Begin	Начинать	Mod	Остаток от деления
Boolean	Логический	Not	Не
Byte	Байт	Or	Или
Char	Символ	Pen	Перо
Color	Цвет	Pixel	Точка
Circle	Круг	Procedure	Процедура
Div	Делить нацело	Program	Программа
Do	Выполнить	Real	Вещественный
Downto	Вниз до ...	Record	Запись
Else	Иначе	Rectangle	Прямоугольник
End	Конец	Set	Установить
File	Файл	String	Строка
For	Для	Then	То
Function	Функция	To	До, к ...
Goto	Идти	Var (variable)	Переменная
If	Если	While	Пока

Таблица 2

**Меню Правка**

Команда	Назначение
Отменить CTRL+Z	Отмена последней операции редактирования текста программы
Восстановить Shift+CTRL+Z	Восстановление предыдущей операции редактирования текста программы
Вырезать CTRL+X	Перемещение выделенного фрагмента текста из окна редактора в буфер обмена
Копировать CTRL+C	Копирование выделенного фрагмента текста из окна редактора в буфер обмена
Вставить CTRL+V	Вставка выделенного текста из буфера обмена в окно редактора
Найти CTRL+F	Поиск текста
Заменить CTRL+R	Поиск текста и замена его новым текстом
Найти далее CTRL+L	Дальнейший поиск текста

Таблица 3

**Меню Файл**

Команда	Назначение
Новый CTRL+N	Открытие окна для нового файла
Открыть CTRL+O	Открытие (загрузка) файла
Сохранить CTRL+S	Сохранение файла с прежним именем
Сохранить как ...	Сохранение файла с новым именем

*Продолжение*

Команда	Назначение
Закрыть CTRL+F4	Закрыть текущее окно
Выход	Выход из системы программирования

Таблица 4

**Сообщения об ошибках при работе в системе Pascal ABC**

Сообщение об ошибке	Возможная причина ошибки
Неожиданный символ	Символ введен не с регистра английских букв
Ожидался символ «точка»	После end нет точки
Ожидалось begin	Нет слова begin
Ожидалась «точка с запятой»	Нет символа «;» либо неверно расставлены скобки в команде
Ожидался идентификатор	Не указано имя программы
Ожидалась команда	Отсутствует end
Неизвестное имя	Неправильно написано название команды
Ожидалось «)»	Не закрыта скобка в тексте команды
Ожидался конец файла	Неверно написано слово end либо есть лишние символы после слова end с точкой в конце программы
Ожидался идентификатор, но ... обнаружено зарезервированное слово	Например, вместо val1 написано var
Ошибка ввода. Программа завершена	Неверно указано имя файла с задачей при ее открытии ('begin')

Таблица 5

**Работа с графическим окном**

Команда	Назначение
SetWindowLeft (l: integer)	Устанавливает отступ графического окна от левого края экрана в пикселях
SetWindowTop (t: integer)	Устанавливает отступ графического окна от верхнего края экрана в пикселях
SetWindowCaption (s: string)	Устанавливает заголовок графического окна
SetWindowTitle (s: string)	Устанавливает заголовок графического окна
SetWindowSize (w,h: integer)	Устанавливает размеры клиентской части графического окна в пикселях
SetWindowPos (l,t: integer)	Устанавливает отступ графического окна от левого верхнего края экрана в пикселях
ClearWindow	Очищает графическое окно белым цветом
ClearWindow(c: Color)	Очищает графическое окно цветом color
SaveWindow (fname: string)	Сохраняет содержимое графического окна в файл с именем fname
LoadWindow (fname: string)	Загружает содержимое графического окна из файла с именем fname

Таблица 6  
Работа с пером

Команда	Назначение
SetPenColor(c: Color)	Устанавливает цвет текущего пера
SetPenWidth(Width: integer)	Устанавливает ширину текущего пера
SetPenStyle(style: DashStyle)	Устанавливает стиль текущего пера
SetPenMode(m: integer)	Устанавливает режим текущего пера

Таблица 7  
Стили пера

Константа	Тип пера
psSolid = DashStyle.Solid;	Сплошное
psDash = DashStyle.Dash;	Штриховое
psDot = DashStyle.Dot;	Пунктирное
psDashDot = DashStyle.DashDot;	Штрихпунктирное
psDashDotDot = DashStyle.DashDotDot;	Альтернативное штрих-пунктирное
psClear = DashStyle.Clear;	Прозрачное

Таблица 8  
Команды модуля GraphABC

Команда	Назначение
SetPixel(x,y,color: integer);	Закрашивает один пиксель с координатами (x,y) цветом color.
MoveTo(x,y: integer);	Передвигает невидимое перо к точке с координатами (x,y); работает в паре с LineTo(x,y).

*Продолжение*

Команда	Назначение
LineTo (x,y: integer);	Рисует отрезок от текущего положения пера до точки (x,y); координаты пера при этом также становятся равными (x,y).
Line(x1,y1,x2,y2: integer);	Рисует отрезок с началом в точке (x1,y1) и концом в точке (x2,y2).
Circle (x,y,r: integer);	Рисует окружность с центром в точке (x,y) и радиусом r.
Ellipse(x1,y1,x2,y2: integer);	Рисует эллипс, заданный своим описанным прямоугольником с координатами противоположных вершин (x1,y1) и (x2,y2).
Rectangle (x1,y1,x2,y2: integer);	Рисует прямоугольник, заданный координатами противоположных вершин (x1,y1) и (x2,y2).
TextOut(x,y: integer; s: string);	Выводит строку s в позицию (x,y) (точка (x,y) задает верхний левый угол прямоугольника, который будет содержать текст из строки s).
FloodFill(x,y,color: integer);	Заливает область одного цвета цветом color начиная с точки (x,y).
FillRect (x1,y1,x2,y2: integer);	Заливает прямоугольник, заданный координатами противоположных вершин (x1,y1) и (x2,y2), цветом текущей кисти.

Таблица 9

**Работа с кистью**

Команда	Назначение
SetBrushColor(c: Color)	Устанавливает цвет текущей кисти
SetBrushStyle(bs: BrushStyleType)	Устанавливает стиль текущей кисти

Таблица 10

**Стили кисти**

Константа	Стиль кисти
bsSolid	Сплошная (по умолчанию)
bsClear	Прозрачная
bsHatch	Штриховая
bsGradient	Градиентная

Таблица 11

**Работа со шрифтом**

Команда	Назначение
SetFontSize(size: integer)	Устанавливает размер текущего шрифта в пикселях
SetFontColor(c: Color)	Устанавливает цвет текущего шрифта
SetFontStyle(fs: integer)	Устанавливает стиль текущего шрифта

# **Содержание**

От авторов .....	3
------------------	---

## **Глава 1**

### **Информация и информационные процессы**

§ 1. Виды информации .....	5
§ 2. Информационные процессы. Носители информации .....	8

## **Глава 2**

### **Аппаратное и программное обеспечение компьютера**

§ 3. Устройства компьютера .....	12
§ 4. Операционная система .....	19
§ 5. Работа с окнами .....	24
§ 6. Файловая система .....	30
§ 7. Операции с объектами файловой системы .....	35
7.1. Файловый менеджер <b>Проводник</b> .....	—
7.2. Создание и переименование папок и файлов .....	38
7.3. Копирование, перемещение и удаление файлов и папок .....	41
7.4. Создание ярлыков .....	45
§ 8. Локальная компьютерная сеть .....	48

## **Глава 3**

### **Основы алгоритмизации и программирования**

§ 9. Среда программирования Pascal ABC .....	54
9.1. Что такое язык программирования? .....	—
9.2. Структура программы .....	55
9.3. Запуск среды Pascal ABC .....	56
9.4. Работа в окне редактора .....	57
9.5. Сохранение и загрузка программ .....	58
9.6. Выполнение программы .....	—
§ 10. Основные понятия языка программирования .....	61
10.1. Понятие переменной .....	—
10.2. Типы данных .....	62
§ 11. Организация ввода и вывода данных .....	63

§ 12. Команда присваивания. Арифметические операции и выражения. Стандартные функции . . . . .	69
12.1. Команда присваивания . . . . .	—
12.2. Работа с переменными целого типа . . . . .	72
12.3. Работа с переменными вещественного типа . . . . .	74
12.4. Преобразование типов числовых данных . . . . .	76
§ 13. Составление линейных программ для решения задач . . . . .	80
§ 14. Графические возможности языка программирования . . . . .	85
14.1. Графический режим работы . . . . .	—
14.2. Возможности стандартной графической библиотеки . . . . .	86
14.3. Рисование линий . . . . .	88
14.4. Рисование прямоугольников и окружностей . . . . .	92

## Глава 4 Компьютерные презентации

§ 15. Понятие компьютерной презентации. Программа PowerPoint . . . . .	98
§ 16. Создание презентации . . . . .	103
§ 17. Вставка изображений. Использование коллекций . .	109
§ 18. Использование декоративного текста . . . . .	113
§ 19. Использование фигур . . . . .	117
§ 20. Элементы мультимедиа . . . . .	120
§ 21. Использование эффектов . . . . .	124
§ 22. Создание презентаций по темам учебных предметов	129

## Глава 5 Компьютерные коммуникации и сеть Интернет

§ 23. Глобальная компьютерная сеть Интернет . . . . .	134
§ 24. Просмотр веб-страниц . . . . .	139
24.1. Элементы интерфейса браузера . . . . .	—
24.2. Работа с браузером . . . . .	142
24.3. Вопросы безопасности в сети Интернет . . . . .	146
Приложение . . . . .	150

Учебное издание  
**Заборовский Георгий Александрович**  
**Козинский Андрей Андреевич**  
**Пупцов Александр Евгеньевич**  
**Якунина Нина Владимировна**

## **ИНФОРМАТИКА**

Учебное пособие для 7 класса  
общеобразовательных учреждений  
с русским языком обучения

Зав. редакцией *В. Г. Бехтина*. Редактор *Н. М. Алганова*. Оформление  
*Ю. В. Прохорчика*. Художественный редактор *Л. А. Дащекевич*. Технический редактор *Е. В. Прудыус*. Корректоры *В. С. Бабеня, Т. Н. Ведерникова, Д. Р. Лосик, А. В. Алешко*.

Подписано в печать 26.06.2009. Формат  $60 \times 90^{1/16}$ . Бумага офсетная.  
Гарнитура школьная. Офсетная печать. Усл.-печ. л. 10. Уч.-изд. л. 6,5.  
Тираж 93 650 экз. Заказ .

Издательское республиканское унитарное предприятие  
«Народная асвета» Министерства информации Республики Беларусь.  
ЛИ 02330/0494083 от 03.02.2009.  
Пр. Победителей, 11, 220004, Минск.

Республиканское унитарное предприятие  
«Минская фабрика цветной печати».  
ЛП № 02330/0494156 от 03.04.2009.  
Ул. Корженевского, 20, 220024, Минск.

**Правообладатель Народная асвета**

(Название и номер школы)

Учебный год	Имя и фамилия ученика	Состояние учебного пособия при получении	Оценка ученику за пользование учебным пособием
20 /			
20 /			
20 /			
20 /			
20 /			