

Федеральное агентство по образованию  
Государственное образовательное учреждение  
высшего профессионального образования  
«Пермский государственный технический университет»

И.Н. Щапова

# **ИНФОРМАТИКА**

Утверждено  
Редакционно-издательским советом университета  
в качестве учебного пособия

Издательство  
Пермского государственного технического университета  
2008

УДК 004(075.8)  
Щ25

Рецензенты:

канд. техн. наук, профессор *Э.В. Любимов*  
(Пермский государственный технический университет);

канд. техн. наук *И.Я. Сальников*  
(ЗАО «Энергосервис»)

**Щапова, И.Н.**

Щ25 Информатика: учеб. пособие / И.Н. Щапова. –  
Пермь: Изд-во Перм. гос. техн. ун-та, 2008. – 115 с.

ISBN 978-5-398-00018-4

Рассмотрены основные вопросы информатики – аппаратное и программное обеспечение компьютера, компьютерные сети, алгоритмизация и программирование. Приведены эффективные приемы работы с программами: операционной системой Windows, текстовым процессором Microsoft Word, электронными таблицами Microsoft Excel, системой управления базой данных Microsoft Access и математическим пакетом Mathcad.

Предназначено для студентов горно-нефтяных специальностей.

УДК 004(075.8)

ISBN 978-5-398-00018-4

© ГОУ ВПО  
«Пермский государственный  
технический университет», 2008

## ОГЛАВЛЕНИЕ

1. Информация и информатика .....	5
1.1. Информатика – предмет и задачи .....	5
1.2. Информация и ее свойства .....	6
1.3. Данные. Операции с данными .....	6
1.4. Основные структуры данных .....	7
2. Кодирование данных двоичным кодом .....	8
2.1. Кодирование целых и действительных чисел .....	9
2.2. Кодирование текстовых данных .....	9
2.3. Кодирование графических данных .....	11
2.4. Кодирование звуковой информации .....	12
3. Состав вычислительной системы .....	13
3.1. Принцип действия компьютера .....	13
3.2. Аппаратное обеспечение .....	13
3.3. Программное обеспечение .....	14
4. Функции операционных систем персональных компьютеров .....	19
4.1. Обеспечение интерфейса пользователя .....	19
4.2. Обеспечение автоматического запуска .....	20
4.3. Организация файловой системы .....	20
4.4. Обслуживание файловой структуры .....	21
4.5. Управление установкой, исполнением и удалением приложений .....	22
4.6. Обеспечение взаимодействия с аппаратными средствами .....	23
4.7. Обслуживание компьютера .....	24
5. Аппаратное обеспечение компьютера .....	24
5.1. Базовая аппаратная конфигурация персонального компьютера .....	24
5.2. Внутренние устройства системного блока .....	25
5.3. Системы, расположенные на материнской плате .....	28
5.4. Периферийные устройства персонального компьютера .....	32
6. Компьютерные сети .....	32
6.1. Основные понятия компьютерных сетей .....	32
6.2. Теоретические основы Интернета .....	34

6.3. Службы Интернета.....	35
6.4. Вопросы компьютерной безопасности.....	41
7. Алгоритмизация и программирование на языке Паскаль...	44
7.1. Алгоритм и его свойства.....	44
7.2. Основные структуры алгоритмов.....	45
7.3. Алгоритмы линейной, разветвляющейся и циклической структуры.....	48
7.4. Программирование на языке Паскаль.....	51
8. Контрольные работы.....	62
8.1. Контрольная работа № 1.....	63
8.2. Контрольная работа № 2.....	65
8.3. Контрольная работа № 3.....	68
Вопросы по дисциплине «Информатика».....	76
Приложение 1. Операционная система Windows XP.....	77
Приложение 2. Текстовый процессор Microsoft Office Word 2003.....	83
Приложение 3. Электронные таблицы Microsoft Office Excel 2003.....	94
Приложение 4. Система управления базой данных Microsoft Access.....	103
Приложение 5. Математический пакет Mathcad.....	108
Список литературы.....	114

# 1. ИНФОРМАЦИЯ И ИНФОРМАТИКА

## 1.1. Информатика – предмет и задачи

*Информатика* – это техническая наука, систематизирующая приемы создания, хранения, воспроизведения, обработки и передачи данных средствами вычислительной техники, а также принципы функционирования этих средств и методы управления ими [1].

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

Предмет информатики составляют следующие понятия:

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

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

Основной задачей информатики является систематизация приемов и методов работы с аппаратными и программными средствами вычислительной техники.

В составе основной задачи информатики можно выделить следующие направления для практических приложений:

- архитектура вычислительных систем (приемы и методы построения систем, предназначенных для автоматической обработки данных);
- интерфейсы вычислительных систем (приемы и методы управления аппаратным и программным обеспечением);
- программирование (приемы, методы и средства разработки компьютерных программ);

- преобразование данных (приемы и методы преобразования структур данных);
- защита информации (обобщение приемов, разработка методов и средств защиты данных);
- автоматизация (функционирование программно-аппаратных средств без участия человека);
- стандартизация (обеспечение совместимости между аппаратными и программными средствами, а также форматами представления данных, относящихся к различным типам вычислительных систем).

## **1.2. Информация и ее свойства**

*Информация* – это продукт взаимодействия данных и адекватных им методов.

С точки зрения информатики наиболее важными являются следующие свойства информации:

- *объективность* (более объективной принято считать ту информацию, в которую методы вносят меньший субъективный элемент);
- *полнота* (означает, что информация содержит минимальный, но достаточный для принятия правильного решения набор показателей);
- *достоверность*;
- *адекватность* (определенный уровень соответствия создаваемого с помощью полученной информации образа реальному объекту, процессу, явлению и т.п.);
- *доступность*;
- *актуальность* (степень соответствия информации текущему моменту времени).

## **1.3. Данные. Операции с данными**

*Данные* – диалектическая составная часть информации. Они представляют собой зарегистрированные сигналы. При этом физический метод регистрации может быть любым (изменение электрических, магнитных, оптических характеристик).

В соответствии с методом регистрации данные могут храниться и транспортироваться на носителях различных видов (бумага, магнитные и оптические диски и т.п.) [1].

Свойства информации весьма тесно связаны со свойствами ее носителей. Любой носитель можно охарактеризовать параметром *разрешающей способности* (количеством данных, записанных в принятой для носителя единице измерения) и *динамическим диапазоном* (логарифмическим отношением интенсивности амплитуд максимального и минимального регистрируемых сигналов). От этих свойств носителя нередко зависят такие свойства информации, как полнота, доступность и достоверность.

В ходе информационного процесса данные преобразуются из одного вида в другой с помощью методов. Обработка данных включает в себя множество различных операций, среди которых можно выделить следующие основные:

- *сбор* данных;
- *формализация* данных (приведение данных, поступающих из разных источников, к одинаковой форме, чтобы сделать их сопоставимыми между собой, т.е. повысить уровень их доступности);
- *фильтрация* данных (отсевание данных, в которых нет необходимости для принятия решения);
- *сортировка* данных (упорядочение данных по заданному признаку с целью удобства их использования);
- *архивация* данных (организация хранения данных в удобной и доступной форме);
- *защита* данных;
- *транспортировка* данных;
- *преобразование* данных (перевод данных из одной формы в другую или из одной структуры в другую).

#### **1.4. Основные структуры данных**

Работа с большими наборами данных автоматизируется проще, когда данные упорядочены, т.е. образуют заданную структуру. Существует три основных типа структур данных: *ли-*

*нейная* – список данных – это упорядоченная структура, в которой адрес элемента однозначно определяется его номером; *табличная* – матрица – это упорядоченная структура, в которой адрес элемента определяется номером строки и номером столбца, на пересечении которых находится ячейка, содержащая искомый элемент; *иерархическая* – это структура, в которой адрес каждого элемента определяется путем доступа (маршрутом), ведущим от вершины структуры к данному элементу.

В качестве единицы хранения данных принят объект переменной длины, называемый *файлом*. Файл – это последовательность произвольного числа байтов, обладающая уникальным собственным именем (тип данных определяет тип файла). Совокупность файлов образует файловую структуру, которая, как правило, относится к иерархическому типу.

## 2. КОДИРОВАНИЕ ДАННЫХ ДВОИЧНЫМ КОДОМ

Для автоматизации работы с данными, относящимися к различным типам, очень важно унифицировать их форму представления. Для этого обычно используется прием кодирования, т.е. выражение данных одного типа через данные другого типа.

Двоичное кодирование, используемое в вычислительной технике, основано на представлении данных последовательностью всего двух знаков: 0 и 1. Эти знаки называются *двоичными цифрами*, по-английски – *binary digit* или сокращенно bit (бит).

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

00    01    10    11

Тремя битами можно закодировать восемь различных значений:

000    001    010    011    100    101    110    111

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



$$N = 2^m,$$

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

## 2.1. Кодирование целых и действительных чисел

Кодирование целых чисел осуществляется следующим образом: число делится пополам до тех пор, пока в остатке не образуется ноль или единица. Совокупность остатков от каждого деления, записанная справа налево, начиная с последнего частного, и образует двоичный аналог десятичного числа.

*Пример.* Перевести  $14_{10} \rightarrow X_2$ .

$14 : 2 = 7$  (остаток **0**),  $7 : 2 = 3$  (остаток **1**),  $3 : 2 = 1$  (остаток **1**).

Таким образом,  $14_{10} \rightarrow 1110_2$ .

Для перевода из двоичной системы счисления в десятичную нужно представить число в виде многочлена и вычислить его значение.

*Пример.* Перевести  $1110_2 \rightarrow X_{10}$ .

$$1110_2 \rightarrow 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8 + 4 + 2 = 14.$$

Таким образом,  $1110_2 \rightarrow 14_{10}$ .

8 разрядов двоичного кода (8 бит) позволяют закодировать целые числа от 0 до 255; 16 бит – целые числа от 0 до 65 535, а 24 бита – более 16,5 млн различных значений.

Для кодирования действительных чисел используют 80-разрядное кодирование. При этом число предварительно преобразуется в нормализованную форму:  $3,1415926 = 0,31415926 \cdot 10^1$ .

Первая часть числа называется мантиссой, а вторая – характеристикой.

## 2.2. Кодирование текстовых данных

Если с каждым символом алфавита сопоставить определенное целое число (например, порядковый номер), то с помощью двоичного кода можно кодировать и текстовую информацию. Восемью двоичных разрядов достаточно для кодирования

256 различных символов. Для того чтобы весь мир одинаково кодировал текстовые данные, нужны единые таблицы кодирования [1].

Институт стандартизации США ввел в действие систему кодирования *ASCII* (American Standard Code for Information Interchange – *стандартный код информационного обмена США*). В системе ASCII закреплены две таблицы кодирования – *базовая* и *расширенная*. Базовая таблица закрепляет значения кодов от 0 до 127, а расширенная относится к символам с номерами от 128 до 255.

Первые 32 кода базовой таблицы, начиная с нулевого, отданы производителям аппаратных средств (в первую очередь производителям компьютеров и печатающих устройств). В этой области размещаются так называемые *управляющие коды*, которым не соответствуют никакие символы.

Начиная с кода 32 по код 127 размещены коды символов английского алфавита, знаков препинания, цифр, арифметических действий и некоторых вспомогательных символов.

Расширенная часть системы кодирования, определяющая значения кодов со 128 по 255, – это национальная система кодирования. Отсутствие единого стандарта в этой области привело к множественности одновременно действующих кодировок. Только в России можно указать три действующих стандарта кодировки: Windows 1251, КОИ-8 (код обмена информацией, восьмизначный), ISO.

Разнообразие кодировок вызвано ограниченным набором кодов (256) при восьмиразрядном кодировании. Универсальная система кодирования UNICODE, основанная на 16-разрядном кодировании символов, позволяет обеспечить уникальные коды для 65 536 различных символов – этого поля достаточно для размещения в одной таблице символов большинства языков планеты. В системе кодирования UNICODE все текстовые документы автоматически становятся вдвое длиннее по сравнению с 8-разрядным кодированием.

### 2.3. Кодирование графических данных

В зависимости от способа формирования изображений компьютерную графику принято подразделять на *растровую* (графический объект представлен в виде комбинации точек, образующих растр и обладающих свойствами яркости и цвета), *векторную* (элементарным объектом является не точка, а линия) и *фрактальную* (базовым элементом является математическая формула). *Трехмерная (3D) графика* сочетает в себе векторный и растровый способ формирования изображений.

Поскольку линейные координаты и индивидуальные свойства каждой точки (яркость) можно выразить с помощью целых чисел, то можно сказать, что растровое кодирование позволяет использовать двоичный код для представления графических данных. Кодирование черно-белых изображений осуществляется восьмиразрядным кодированием, что позволяет отобразить 256 градаций серого цвета.

Для кодирования цветных графических изображений применяется *принцип декомпозиции* произвольного цвета на основные составляющие. В качестве таких составляющих используют три основных цвета: красный (Red, R), зеленый (Green, G) и синий (Blue, B). Такая система кодирования называется системой RGB. Если для кодирования яркости каждой из основных составляющих использовать по 256 значений (8 двоичных разрядов), то на кодирование цвета одной точки надо затратить 24 разряда, что обеспечивает определение 16,5 млн различных цветов. Такой режим называется *полноцветным (True Color)*.

Каждому из основных цветов можно поставить в соответствие дополнительный цвет, то есть цвет, дополняющий основной цвет до белого. Для любого из основных цветов дополнительным будет цвет, образованный суммой пары остальных основных цветов. Соответственно, дополнительными цветами являются: голубой (Cyan, C), пурпурный (Magenta, M) и желтый (Yellow, Y). Принцип декомпозиции произвольного цвета на составляющие компоненты можно применять не только для основных цветов, но и для дополнительных, то есть любой цвет можно представить в виде суммы голубой, пурпурной и желтой

составляющих. Такой метод кодирования цвета с использованием четвертой краски – черной (Black, K) – принят в полиграфии. Данная система кодирования обозначается четырьмя буквами CMYK и для представления цветной графики в этой системе надо иметь 32 двоичных разряда. Такой режим тоже называется *полноцветным (True Color)*.

## **2.4. Кодирование звуковой информации**

В кодировании звуковой информации можно выделить два основных направления.

Метод FM (Frequency Modulation) основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических сигналов разных частот, каждый из которых представляет собой правильную синусоиду, а следовательно, может быть описан числовыми параметрами, т.е. кодом. Разложение звуковых сигналов в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства – аналого-цифровые преобразователи (АЦП). Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют цифроаналоговые преобразователи (ЦАП). При таких преобразованиях неизбежны потери информации, поэтому качество звукозаписи не вполне удовлетворительное. В то же время данный метод кодирования обеспечивает весьма компактный код.

Метод таблично-волнового (Wave-Table) синтеза лучше соответствует современному уровню развития техники. Проще говоря, где-то в заранее подготовленных таблицах хранятся образцы звуков (сэмплы) для множества различных музыкальных инструментов. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения и другие показатели. Качество звука, полученного в результате синтеза, получается очень высокое.

## 3. СОСТАВ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

### 3.1. Принцип действия компьютера

В основе любого современного компьютера лежит *тактовый генератор*, вырабатывающий через равные интервалы времени электрические сигналы, которые используются для приведения в действие всех устройств компьютерной системы. В ПК тактовые импульсы задает одна из микросхем, входящая в микропроцессорный комплект (чипсет), расположенный на материнской плате. Чем выше частота тактов, поступающих на процессор, тем больше команд он может исполнить в единицу времени (исполнение каждой команды занимает определенное количество тактов) и тем выше его производительность [1].

Управление компьютером фактически сводится к управлению распределением сигналов между устройствами. Такое управление может производиться автоматически (*программное управление*). В современных компьютерах внешнее управление в значительной степени автоматизировано с помощью специальных аппаратно-логических интерфейсов, к которым подключаются устройства управления и ввода данных (клавиатура, мышь, джойстик и др.). Такое управление называется *интерактивным*.

Состав вычислительной системы называется *конфигурацией*. Отдельно рассматривают *аппаратную конфигурацию* вычислительных систем и их *программную конфигурацию*.

Несмотря на то, что программное и аппаратное обеспечение рассматриваются отдельно, в компьютере они работают в неразрывной связи и при непрерывном взаимодействии.

### 3.2. Аппаратное обеспечение

*Аппаратная конфигурация* представляет собой блочно-модульную конструкцию, т.е. конфигурацию, состоящую из готовых узлов и блоков.

По способу расположения устройств относительно *центрального процессорного устройства (ЦПУ, CPU)* различают *внутренние* и *внешние* устройства. *Внешними*, как правило, яв-

ляются большинство устройств ввода-вывода данных (периферийные устройства) и некоторые устройства длительного хранения данных.

Согласование между отдельными узлами и блоками выполняют с помощью переходных аппаратно-логических устройств, называемых *аппаратными интерфейсами*. Стандарты на аппаратные интерфейсы называют *протоколами*. Таким образом, *протокол* – это совокупность технических условий, которые должны быть обеспечены разработчиками устройств для успешного согласования их работы с другими устройствами.

Интерфейсы в архитектуре вычислительной системы можно условно разделить на две большие группы: *последовательные* и *параллельные*. Через последовательный интерфейс данные передаются бит за битом, а через параллельный – одновременно группами битов (количество битов, участвующих в одной передаче, определяется разрядностью интерфейса). Параллельные интерфейсы обычно имеют более сложное устройство, чем последовательные, но обеспечивают более высокую производительность. Для параллельных интерфейсов ее измеряют байтами в секунду (байт/с; кбайт/с; Мбайт/с).

Устройство последовательных интерфейсов проще, как правило, для них не надо синхронизировать работу передающего и принимающего устройств (поэтому их часто называют асинхронными интерфейсами), но пропускная способность их меньше и коэффициент полезного действия ниже, так как из-за отсутствия синхронизации посылки полезные данные предваряют и завершают посылками служебных данных. Поэтому производительность последовательных устройств измеряют битами в секунду (бит/с; кбит/с; Мбит/с).

### 3.3. Программное обеспечение

*Программы* – это упорядоченные последовательности команд. Конечная цель любой компьютерной программы – управление аппаратными средствами.

Состав программного обеспечения (ПО) вычислительной системы называют *программной конфигурацией*. Между программами, как и между физическими узлами и блоками,

существует взаимосвязь – многие программы работают, опираясь на другие программы более низкого уровня (*межпрограммный интерфейс*). Таким образом, программное обеспечение можно распределить на несколько взаимодействующих между собой уровней, представляющих пирамидальную конструкцию (рис. 1). Каждый



Рис. 1. Уровни программного обеспечения

следующий уровень опирается на программное обеспечение предшествующих уровней, и каждый вышележащий уровень повышает функциональность всей системы [1].

**Базовый уровень.** Отвечает за взаимодействие с базовыми аппаратными средствами. Как правило, базовые программные средства непосредственно входят в состав базового оборудования и хранятся в специальных микросхемах, называемых *постоянными запоминающими устройствами (ПЗУ – Read Only Memory, ROM)*. Программы и данные записываются («прошиваются») в микросхемы ПЗУ на этапе производства и не могут быть изменены в процессе эксплуатации.

В тех случаях, когда изменение базовых программных средств во время эксплуатации является технически целесообразным, вместо микросхем ПЗУ применяют *перепрограммируемые постоянные запоминающие устройства*.

**Системный уровень.** Программы, работающие на этом уровне, обеспечивают взаимодействие прочих программ компьютерной системы с программами базового уровня и непосредственно с аппаратным обеспечением.

В состав программного обеспечения системного уровня входят программы, отвечающие за взаимодействие с конкретными устройствами, называемые *драйверами устройств*.

Другой класс программ системного уровня отвечает за взаимодействие с пользователем (средства обеспечения пользовательского интерфейса).

Совокупность программного обеспечения системного уровня образует *ядро операционной системы* компьютера, например, операционная система Windows (приложение 1).

**Служебный уровень.** Программное обеспечение этого уровня взаимодействует как с программами базового уровня, так и с программами системного уровня. Основное назначение служебных программ (их также называют *утилитами*) состоит в автоматизации работ по проверке, наладке и настройке компьютерной системы. Во многих случаях они используются для расширения или улучшения функций системных программ.

***Классификация служебных программных средств:***

– *Диспетчеры файлов (файловые менеджеры)*. С их помощью выполняется большинство операций, связанных с обслуживанием файловой структуры: копирование, перемещение и переименование файлов, создание и удаление файлов и каталогов, навигация в файловой структуре (например, программа *Проводник* – стандартное приложение операционной системы Windows).

– *Средства сжатия данных (архиваторы)*. Предназначены для создания архивов с целью повышения эффективности использования носителя за счет того, что архивные файлы обычно имеют повышенную плотность записи информации.

– *Средства просмотра и воспроизведения*.

– *Средства диагностики*. Предназначены для автоматизации процессов диагностики программного и аппаратного обеспечения.

– *Средства контроля (мониторинга)*. Позволяют следить за процессами, происходящими в компьютерной системе.

– *Мониторы установки*. Предназначены для контроля над установкой программного обеспечения (для протоколирования образования связей между различными категориями программного обеспечения).

– *Средства коммуникации*. Позволяют устанавливать соединения с удаленными компьютерами, обслуживают передачу сообщений электронной почты, работу с телеконференциями (группами новостей) и др.



– Средства обеспечения компьютерной безопасности.

К ним относятся средства пассивной защиты (программы для резервного копирования) и активной защиты (антивирусное программное обеспечение, например программы NOD32, Dr.Web, Антивирус Касперского 7.0), а также средства защиты от несанкционированного доступа, просмотра и изменения данных.

**Прикладной уровень.** Представляет собой комплекс прикладных программ, с помощью которых выполняются конкретные задания.

**Классификация прикладных программных средств:**

– *Текстовые редакторы.* Обеспечивают ввод и редактирование текстовых данных.

– *Текстовые процессоры.* Позволяют не только вводить и редактировать текст, но и форматировать его, т.е. оформлять, например, Microsoft Word (приложение 2).

– *Графические редакторы.* Предназначены для создания и обработки графических изображений. Различают следующие категории графических редакторов: *растровые редакторы* (например, Adobe Photoshop), *векторные редакторы* (например, CorelDraw) и программные средства для создания и обработки трехмерной графики (*3D-редакторы*, например, 3D Studio Max).

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

Векторные редакторы отличаются от растровых способом представления данных об изображении. Элементарным объектом векторного изображения является не точка, а линия. Такой подход характерен для чертежно-графических работ, в которых форма линий имеет большее значение, чем информация о цвете отдельных точек, составляющих ее.

– *Электронные таблицы.* Предоставляют комплексные средства для хранения различных типов данных и их обработ-

ки, например Microsoft Excel (приложение 3). Основное свойство электронных таблиц состоит в том, что при изменении содержания любых ячеек таблицы может происходить автоматическое изменение содержания во всех прочих ячейках, связанных с измененными соотношениями, заданными математическими или логическими выражениями (формулами). Применяются там, где необходимо автоматизировать регулярно повторяющиеся вычисления достаточно больших объемов числовых данных.

– *Системы управления базами данных (СУБД)*. Базами данных называют массивы данных, организованных в табличные структуры. Основными функциями СУБД, например Microsoft Access (приложение 4), являются:

1) создание пустой (незаполненной) структуры базы данных;

2) предоставление средств ее заполнения или импорта данных из таблиц другой базы;

3) обеспечение возможности доступа к данным, а также предоставление средств поиска и фильтрации.

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

– *Специальные пакеты (математические, статистические и др.)*. Обеспечивают решение задач в определенной предметной области, например в математике (программа Mathcad) (приложение 5), статистике, банковском деле и т.д.

– *Экспертные системы*. Предназначены для анализа данных, содержащихся в базах знаний, и выдачи рекомендаций по запросу пользователя. Такие системы применяют в тех случаях, когда исходные данные хорошо формализуются, но для принятия решений требуются обширные специальные знания (юриспруденция, медицина, химия).

– *Web-редакторы*. Предназначены для создания и редактирования Web-документов (Web-страниц Интернета). Объеди-

няют в себе свойства текстовых и графических редакторов (например, программа FrontPage из пакета Microsoft Office).

– *Браузеры (обозреватели, средства просмотра Web)*. Программные средства, предназначенные для просмотра электронных документов, выполненных в формате HTML (например, Microsoft Internet Explorer).

– *Интегрированные системы делопроизводства*. Представляют собой программные средства автоматизации рабочего места руководителя.

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

#### **4. ФУНКЦИИ ОПЕРАЦИОННЫХ СИСТЕМ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ**

*Операционная система* представляет комплекс системных и служебных программных средств. *Приложениями операционной системы* называют программы, предназначенные для работы под управлением данной системы.

Основная функция операционных систем – посредническая [1]. Она заключается в обеспечении нескольких видов интерфейса:

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

##### **4.1. Обеспечение интерфейса пользователя**

По реализации интерфейса пользователя различают *неграфические* и *графические* операционные системы. Неграфические операционные системы (например, MS-DOS) реализуют *интерфейс командной строки*. Основным устройством управления в этом случае является клавиатура.

Работа с графической операционной системой основана на взаимодействии активных (указатель мыши) и пассивных (строки меню, экранные кнопки, значки, раскрывающиеся списки и т.д.) экранных элементов управления.

#### **4.2. Обеспечение автоматического запуска**

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

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

#### **4.3. Организация файловой системы**

Все современные дисковые операционные системы обеспечивают создание файловой системы, предназначенной для хранения данных на дисках и обеспечения доступа к ним. Принцип организации файловой системы – табличный. Поверхность жесткого диска рассматривается как трехмерная матрица, измерениями которой являются номера поверхности, цилиндра и сектора. Под цилиндром понимается совокупность всех дорожек, принадлежащих разным поверхностям и находящихся на равном удалении от оси вращения. Данные о том, в каком месте диска записан тот или иной файл, хранятся в системной области диска в специальных *таблицах размещения файлов* (FAT-таблицах).

Наименьшей физической единицей хранения данных является сектор. Размер сектора равен 512 байт. Поскольку размер FAT-таблицы ограничен, то для дисков, размер которых превышает 32 Мбайт, обеспечить адресацию к каждому отдельному сектору не представляется возможным. В связи с этим группы секторов условно объединяются в кластеры. Кластер является наименьшей единицей адресации к данным. Размер кластера, в отличие от размера сектора, не фиксирован и зависит от емкости диска. Для дисков размером до 8 Гбайт FAT32 (файловая система с 32-разрядными полями в таблице размещения файлов) обеспечивает размер кластера 4 кбайт (8 секторов). В файловой системе NTFS размер кластера не зависит от размера диска, и, потенциально, для очень больших дисков эта система должна работать эффективнее, чем FAT32.

#### **4.4. Обслуживание файловой структуры**

Несмотря на то, что данные о местоположении файлов хранятся в табличной структуре, пользователю они представляются в виде иерархической структуры (файловой). К функции обслуживания файловой структуры относятся следующие операции, происходящие под управлением операционной системы:

- создание файлов и присвоение им имен;
- создание каталогов (папок) и присвоение им имен;
- переименование файлов и каталогов (папок);
- копирование и перемещение файлов;
- удаление файлов и каталогов (папок) (существует как минимум три режима удаления данных: удаление, уничтожение и стирание);
- навигация по файловой структуре с целью доступа к файлу, каталогу (папке);
- управление атрибутами файлов (кроме имени и расширения имени файла операционная система хранит для каждого файла дату его создания или изменения и несколько величин, называемых атрибутами файла, например, «только для чтения», «скрытый», «системный», «архивный»). Операционная система позволяет их контролировать и изменять, состояние атри-

бутов учитывается при проведении автоматических операций с файлами.

#### **4.5. Управление установкой, исполнением и удалением приложений**

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

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

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

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

Для правильной работы приложений на компьютере они должны пройти операцию, называемую *установкой*. Необходимость в установке связана с тем, что разработчики программно-

го обеспечения не могут заранее предвидеть особенности аппаратной и программной конфигурации вычислительной системы, на которой предстоит работать их программам. Таким образом, *дистрибутивный комплект (установочный пакет)* программного обеспечения, как правило, представляет собой незаконченный программный продукт, из которого в процессе установки на компьютере формируется полноценное рабочее приложение. При этом осуществляется привязка приложения к существующей аппаратно-программной среде и его настройка на работу именно в этой среде. Управление установкой приложений выполняют операционные системы. Они управляют распределением ресурсов вычислительной системы между приложениями, обеспечивают доступ устанавливаемых приложений к драйверам устройств, формируют общие ресурсы, которые могут использоваться разными приложениями, выполняют регистрацию установленных приложений и выделенных им ресурсов.

Процесс удаления приложений имеет свои особенности и происходит под строгим контролем операционной системы. Нельзя допустить, чтобы при удалении одного приложения были удалены ресурсы, на которые опираются другие приложения, даже если эти ресурсы были установлены вместе с удаляемым приложением.

#### **4.6. Обеспечение взаимодействия с аппаратными средствами**

Средства аппаратного обеспечения вычислительной техники отличаются огромным многообразием. Гибкость аппаратных и программных конфигураций вычислительных систем поддерживается за счет того, что каждый разработчик оборудования прикладывает к нему специальные программные средства управления – драйверы. Драйверы имеют точки входа для взаимодействия с прикладными программами, а диспетчеризация обращений прикладных программ к драйверам устройств – это одна из функций операционной системы. Операционная система выполняет все функции по установке драйверов устройств и передаче им управления от приложений. Во многих случаях операционная система даже не нуждается в драйверах, полученных

от разработчиков устройств, а использует драйверы из собственной базы данных.

Современные операционные системы позволяют управлять не только установкой и регистрацией программных драйверов устройств, но и процессом аппаратно-логического подключения, при этом реализуется принцип динамического распределения ресурсов операционной системой, который называется *plug-and-play*, а устройства, удовлетворяющие этому принципу, называются *самоустанавливающимися*.

#### **4.7. Обслуживание компьютера**

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

### **5. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА**

#### **5.1. Базовая аппаратная конфигурация персонального компьютера**

Конфигурацию (состав оборудования) ПК можно гибко изменять по мере необходимости. Тем не менее существует понятие *базовой конфигурации*, которую считают типовой. Она включает в себя:

- системный блок;
- монитор;
- клавиатуру (ее основные функции не нуждаются в поддержке специальными системными программами (драйверами) – необходимое программное обеспечение имеется в ПЗУ в составе базовой системы ввода/вывода BIOS);
- мышь (нуждается в поддержке специальной системной программы – драйвера мыши, поэтому в первый момент после включения компьютера мышь не работает).

Основные потребительские параметры монитора:

- размер экрана;



– шаг маски экрана (для монитора на основе электронно-лучевой трубки) и зернистость изображения (для жидкокристаллического монитора);

– максимальная частота регенерации изображения (показывает, сколько раз в течение секунды монитор может полностью сменить изображение);

– класс защиты (определяется стандартом, которому соответствует монитор с точки зрения требований техники безопасности).

Большинством параметров изображения на экране монитора (экраным разрешением, цветовым разрешением и др.) можно управлять программно.

## 5.2. Внутренние устройства системного блока

**Материнская плата.** На материнской плате размещаются:

– *процессор* – основная микросхема, выполняющая большинство математических и логических операций;

– *микروпроцессорный комплект – чипсет* – набор микросхем, управляющих работой внутренних устройств компьютера и определяющих основные функциональные возможности материнской платы;

– *шины* – наборы проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера;

– *оперативная память (оперативное запоминающее устройство, ОЗУ)* – набор микросхем, предназначенных для временного хранения данных, когда компьютер включен;

– *ПЗУ (постоянное запоминающее устройство)* – микросхема, предназначенная для длительного хранения данных;

– разъемы для подключения дополнительных устройств (*слоты*).

**Жесткий диск.** Это основное устройство для долговременного хранения больших объемов данных и программ.

Управление работой жесткого диска выполняет специальное аппаратно-логическое устройство – контроллер жесткого диска. В настоящее время функции контроллеров дисков выпол-

няют микросхемы, входящие в микропроцессорный комплект (чипсет).

К основным параметрам жестких дисков относятся *емкость* (зависит от технологии изготовления) и *производительность* (зависит от характеристик интерфейса, с помощью которого они связаны с материнской платой). С производительностью диска напрямую связан параметр *среднего времени доступа*. Он определяет интервал времени, необходимый для поиска нужных данных, и зависит от скорости вращения диска.

**Дисковод гибких дисков.** Для оперативного переноса небольших объемов информации используют так называемые *гибкие магнитные диски* (дискеты), которые вставляют в специальный накопитель – дисковод.

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

**Дисковод компакт-дисков CD-ROM.** Принцип действия этого устройства состоит в считывании числовых данных с помощью лазерного луча, отражающегося от поверхности диска. Цифровая запись на компакт-диске отличается от записи на магнитных дисках очень высокой плотностью, и стандартный компакт-диск может хранить около 700 Мбайт данных.

Основным недостатком стандартных дисководов CD-ROM является невозможность записи данных, но существуют и устройства записи компакт-дисков – дисководы CD-RW. Для записи используются специальные заготовки. Некоторые из них допускают только однократную запись, другие позволяют стереть ранее записанную информацию и выполнить запись заново.

Основными параметрами дисководов компакт-дисков являются скорость чтения и скорость записи. Они измеряются в кратных долях. За единицу измерения принята скорость чтения музыкальных компакт-дисков, составляющая 150 кбайт/с. Таким образом, дисковод с удвоенной скоростью обеспечивает производительность 300 кбайт/с и т.д.

**DVD** – носитель информации в виде диска, внешне схожий с компакт-диском, однако имеющий возможность хранить

больший объем информации (более 4,7 Гбайт) за счет использования лазера с меньшей длиной волны, чем для обычных компакт-дисков.

DVD как носители бывают четырех типов:

- DVD-ROM – диски только для считывания информации;
- DVD+R/RW – диски однократной (R – Recordable) и многократной (RW – ReWritable) записи;
- DVD-R/RW – диски однократной (R – Recordable) и многократной (RW – ReWritable) записи;
- DVD-RAM – диски многократной записи с произвольным доступом (RAM – Random Access Memory).

Второй и третий типы DVD («+» и «-») отличаются стандартом записи.

**Видеокарта (видеоадаптер).** Совместно с монитором видеокарта образует видеоподсистему персонального компьютера. Физически видеоадаптер выполнен в виде отдельной дочерней платы, которая вставляется в один из слотов материнской платы и называется видеокартой. Видеоадаптер выполняет функции *видеоконтроллера*, *видеопроцессора* и *видеопамяти*.

Основные параметры видеоподсистемы:

– *разрешение экрана* (количество точек по горизонтали и вертикали). Для каждого размера монитора существует свое оптимальное разрешение экрана, которое должен обеспечивать видеоадаптер;

– *цветовое разрешение (глубина цвета)*. Определяет количество различных оттенков, которые может принимать отдельная точка экрана. Максимально возможное цветовое разрешение зависит от свойств видеоадаптера и, в первую очередь, от количества установленной на нем видеопамяти. Кроме того, оно зависит и от установленного разрешения экрана;

– *видеоускорение* – одно из свойств видеоадаптера, которое заключается в том, что часть операций по построению изображений может происходить без выполнения математических вычислений в основном процессоре компьютера, а чисто аппаратным путем – преобразованием данных в микросхемах видеоускорителя.

**Звуковая карта.** Устанавливается в один из разъемов материнской платы в виде дочерней карты и выполняет вычислительные операции, связанные с обработкой звука, речи, музыки.

Основным параметром звуковой карты является *разрядность*, определяющая количество битов, используемых при преобразовании сигналов из аналоговой в цифровую форму и наоборот. Чем выше разрядность, тем меньше погрешность, связанная с оцифровкой, тем выше качество звучания.

При отсутствии повышенных требований к качеству звука можно использовать интегрированные звуковые системы, в которых функции обработки звука выполняются центральным процессором и микросхемами материнской платы.

### 5.3. Системы, расположенные на материнской плате

**Оперативная память.** *Оперативная память (RAM – Random Access Memory)* – это массив кристаллических ячеек, способных хранить данные. С точки зрения физического принципа действия различают *динамическую память (DRAM)* и *статическую память (SRAM)*.

*Ячейки динамической памяти (DRAM)* можно представить в виде микроконденсаторов, способных накапливать заряд на своих обкладках. Каждый бит такой памяти представляется в виде наличия (или отсутствия) заряда, именно этим и объясняется энергозависимость оперативной памяти, т.е. потеря информации при выключении компьютера. Это наиболее распространенный и экономически доступный тип памяти. Недостатки этого типа связаны, во-первых, с тем, что как при заряде, так и при разряде конденсаторов неизбежны переходные процессы, т.е. запись данных происходит сравнительно медленно. Вторым важным недостатком связан с тем, что заряды ячеек имеют свойство рассеиваться в пространстве. Если оперативную память постоянно не «подзаряжать», утрата данных происходит через несколько сотых долей секунды. Чтобы избежать этого, в компьютере несколько десятков раз в секунду происходит регенерация (подзарядка) ячеек оперативной памяти. Это вызывает непроизводительный расход ресурсов вычислительной системы.

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

Микросхемы динамической памяти используют в качестве основной оперативной памяти компьютера. Микросхемы статической памяти используют в качестве вспомогательной памяти (кеш-памяти), предназначенной для оптимизации работы процессора, а именно для согласования работы сравнительно медленных устройств (в этом случае это DRAM) со сравнительно быстрым микропроцессором.

Каждая ячейка памяти имеет свой адрес, который выражается числом. Предельный размер поля оперативной памяти, установленной в компьютере, определяется микропроцессорным комплектом (чипсетом) материнской платы.

Оперативная память в компьютере размещается на стандартных панельках, называемых *модулями*, которые вставляют в соответствующие разъемы на материнской плате. На компьютерах применяются модули памяти DDR SDRAM (DDR DIMM-модули) и RDRAM (RIMM-модули).

Основными характеристиками модулей оперативной памяти являются *объем памяти* и *скорость передачи данных*. Скорость передачи данных определяет максимальную пропускную способность памяти (в Мбайт/с или Гбайт/с) в оптимальном режиме доступа. Одинаковые по объему модули могут иметь разные скоростные характеристики. Иногда в качестве определяющей характеристики памяти используют *время доступа*. Оно измеряется в миллиардных долях секунды (*наносекундах, нс*).

**Процессор.** Процессор – основная микросхема компьютера, в которой и производятся все вычисления. Конструктивно процессор состоит из ячеек, похожих на ячейки оперативной памяти, но в этих ячейках данные могут не только храниться, но и изменяться. Внутренние ячейки процессора называют *регистрами*. Таким образом, любой процессор состоит из набора реги-

стров различного назначения (данные, адресные данные, команды), которые определенным образом связаны между собой и обрабатываются в соответствии с некоторой системой правил. Управляя засылкой данных в разные регистры процессора, можно управлять обработкой данных. На этом и основано исполнение программ.

К обязательным компонентам процессора относятся *арифметико-логическое устройство* и *устройство управления*. Выполнение процессором команды предусматривает: арифметические действия, логические операции, передачу управления, перемещение данных из одного места памяти в другое и координацию взаимодействия различных устройств компьютера. Выделяют 4 этапа обработки команды процессором: выборка команды из ОЗУ, декодирование, выполнение и запись результата.

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

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

***Шина данных.*** По этой шине происходит копирование данных из оперативной памяти в регистры процессора и обратно. По 64-разрядной шине данных за один раз на обработку поступают сразу 8 байт.

***Шина команд.*** Для того чтобы процессор мог обрабатывать данные, ему нужны команды. Эти команды поступают в процессор из той области оперативной памяти, где хранятся программы. Команды тоже представлены в виде байтов. В большинстве процессоров шина команд 32-разрядная, хотя существуют 64-разрядные процессоры и даже 128-разрядные.

***Основные параметры процессоров.*** Основными параметрами процессоров являются: *рабочее напряжение*, *разрядность*,

*рабочая тактовая частота, коэффициент внутреннего умножения тактовой частоты и размер кеш-памяти.*

*Рабочее напряжение* процессора обеспечивает материнская плата. По мере развития процессорной техники происходит постепенное понижение рабочего напряжения. В настоящее время оно составляет менее 3 В. Понижение рабочего напряжения позволяет уменьшить расстояния между структурными элементами в кристалле процессора до десятитысячных долей миллиметра, не опасаясь электрического пробоя. Пропорционально квадрату напряжения уменьшается и тепловыделение в процессоре, а это позволяет увеличивать его производительность без угрозы перегрева.

*Разрядность процессора* показывает, сколько бит данных он может принять и обработать в своих регистрах за один раз (за один такт). При этом разрядность процессора определяется разрядностью не шины данных, а командной шины.

Исполнение каждой команды процессором занимает определенное количество тактов. Чем выше частота тактов, поступающих на процессор, тем больше команд он может исполнить в единицу времени, тем выше его производительность. В настоящее время *рабочие частоты* процессоров превосходят миллиард тактов в секунду (1 ГГц).

Тактовые сигналы процессор получает от материнской платы, которая представляет собой большой набор проводников и микросхем и по чисто физическим причинам не может работать со столь высокими частотами, как процессор. Для получения более высоких частот в процессоре происходит *внутреннее умножение частоты* на коэффициент.

Обмен данными внутри процессора происходит в несколько раз быстрее, чем обмен с другими устройствами, например с оперативной памятью. Для того чтобы уменьшить количество обращений к оперативной памяти, внутри процессора создают буферную область – *кеш-память*. Когда процессору нужны данные, он сначала обращается в кеш-память, и только если там нужных данных нет, происходит его обращение в оперативную память. Принимая блок данных из оперативной памяти, процессор заносит его одновременно и в кеш-

память. Нередко кеш-память распределяют по нескольким уровням.

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

#### **5.4. Периферийные устройства персонального компьютера**

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

По назначению периферийные устройства можно подразделить:

- на устройства ввода данных (специальные манипуляторы, сканеры, цифровые фотокамеры);
- устройства вывода данных (принтеры);
- устройства хранения данных (магнитооптические устройства, флеш-диски и др.);
- устройства обмена данными (модемы).

### **6. КОМПЬЮТЕРНЫЕ СЕТИ**

#### **6.1. Основные понятия компьютерных сетей**

При физическом соединении двух или более компьютеров образуется компьютерная сеть. В общем случае для создания компьютерных сетей необходимо специальное аппаратное обеспечение (сетевое оборудование) и специальное программное обеспечение (сетевые программные средства) [1].

Основной задачей, решаемой при создании компьютерных сетей, является обеспечение совместимости оборудования по электрическим и механическим характеристикам и информационного обеспечения (программ и данных) по системе кодирова-



ния и формату данных. Решение этой задачи относится к области стандартизации и основано на так называемой модели OSI (модель взаимодействия открытых систем). Она создана на основе технических предложений Международного института стандартизации ISO.

Согласно модели ISO/OSI архитектуру компьютерных сетей следует рассматривать на разных уровнях (общее число уровней – до семи). Самый верхний уровень – *прикладной*. На этом уровне пользователь взаимодействует с вычислительной системой. Самый нижний уровень – *физический*. Он обеспечивает обмен сигналами между устройствами. Обмен данными в системах связи происходит путем их перемещения с верхнего уровня на нижний, затем транспортировки и, наконец, обратным воспроизведением на компьютере клиента в результате перемещения с нижнего уровня на верхний.

Для обеспечения необходимой совместимости на каждом из семи возможных уровней действуют специальные стандарты, называемые *протоколами*. Они определяют характер аппаратного взаимодействия компонентов сети (*аппаратные протоколы*) и характер взаимодействия программ и данных (*программные протоколы*).

В соответствии с используемыми протоколами компьютерные сети подразделяют на *локальные (LAN – Local Area Network)* и *глобальные (WAN – Wide Area Network)*. Компьютеры локальной сети преимущественно используют единый комплект протоколов для всех участников. По территориальному признаку локальные сети отличаются компактностью. Глобальные сети имеют, как правило, увеличенные географические размеры. Они могут объединять как отдельные компьютеры, так и отдельные локальные сети, в том числе и использующие различные протоколы.

Назначение всех видов компьютерных сетей определяется двумя функциями:

- обеспечение *совместного использования* аппаратных и программных ресурсов сети;
- обеспечение *совместного доступа* к ресурсам данных.

Уровни модели связи ISO/OSI:

1. На *прикладном уровне* с помощью специальных приложений пользователь создает документ.

2. На *уровне представления* операционная система его компьютера фиксирует, где находятся созданные данные (в оперативной памяти, в файле на жестком диске и т.п.), и обеспечивает взаимодействие со следующим уровнем.

3. На *сеансовом уровне* компьютер пользователя взаимодействует с локальной или глобальной сетью. Протоколы этого уровня проверяют права пользователя на «выход в эфир» и передают документ к протоколам транспортного уровня.

4. На *транспортном уровне* документ преобразуется в ту форму, в которой положено передавать данные в используемой сети. Например, он может нарезаться на небольшие пакеты стандартного размера.

5. *Сетевой уровень* определяет маршрут движения данных в сети. Например, каждый пакет получает адрес, по которому он должен быть доставлен независимо от прочих пакетов.

6. *Уровень соединения* необходим для того, чтобы промодулировать сигналы, циркулирующие на физическом уровне, в соответствии с данными, полученными с сетевого уровня (эти функции в компьютере выполняет сетевая карта или модем).

7. Реальная передача данных происходит на *физическом уровне*. Здесь нет ни документов, ни пакетов, ни даже байтов – только биты, т.е. элементарные единицы представления данных. Восстановление документа из них произойдет при переходе с нижнего на верхний уровень на компьютере клиента.

## 6.2. Теоретические основы Интернета

По-настоящему рождением Интернета принято считать 1983 год. В этом году произошли революционные изменения в программном обеспечении компьютерной связи. Днем рождения Интернета в современном понимании этого слова стала дата стандартизации протокола связи TCP/IP, лежащего в основе Всемирной сети по нынешний день.

TCP/IP – это не один сетевой протокол, а два протокола, лежащих на разных уровнях. Протокол TCP – протокол *транс-*

*портного уровня*. Он управляет тем, как происходит передача информации. Протокол IP – *адресный*. Он принадлежит *сетевому уровню* и определяет, куда происходит передача.

Согласно протоколу TCP отправляемые данные «нарезаются» на небольшие пакеты, после чего каждый пакет маркируется таким образом, чтобы в нем были данные, необходимые для правильной сборки документа на компьютере получателя. Два компьютера, связанных между собой одним физическим соединением, могут поддерживать одновременно несколько TCP-соединений. Так, например, два промежуточных сетевых сервера могут одновременно по одной линии связи передавать друг другу в обе стороны множество TCP-пакетов от многочисленных клиентов.

Суть адресного протокола – IP (Internet Protocol) – состоит в том, что у каждого участника Всемирной сети должен быть свой уникальный адрес (IP-адрес). Этот адрес выражается четырьмя байтами, например: 195.38.46.11. Поскольку один байт содержит до 256 различных значений, то теоретически с помощью четырех байтов можно выразить более четырех миллиардов уникальных IP-адресов.

### **6.3. Службы Интернета**

Когда говорят о работе в Интернете или об использовании Интернета, то на самом деле речь идет не об Интернете в целом, а только об одной или нескольких из его многочисленных служб.

Разные службы имеют разные протоколы. Они называются *прикладными протоколами*. Их соблюдение обеспечивается и поддерживается работой специальных программ. Таким образом, чтобы воспользоваться какой-то из служб Интернета, необходимо установить на компьютере программу, способную работать по протоколу данной службы. Такие программы называют *клиентскими* или просто *клиентами*.

Так, например, чтобы воспользоваться электронной почтой, необходимо соблюсти протоколы отправки и получения сообщений. Для этого надо иметь программу (*почтовый клиент*) и установить связь с *почтовым сервером*.

**Терминальный режим.** Исторически одной из ранних является служба удаленного управления компьютером *Telnet*. Подключившись к удаленному компьютеру по протоколу этой службы, можно управлять его работой. Часто протоколы *Telnet* применяют для дистанционного управления техническими объектами, например телескопами, видеокамерами, промышленными роботами.

Сервер, предоставляющий *Telnet*-услуги, обычно предлагает свое клиентское приложение. Его надо получить по сети, установить на своем компьютере, подключиться к серверу и работать с удаленным оборудованием.

**Электронная почта (E-Mail).** Эта служба также является одной из наиболее ранних. Ее обеспечением в Интернете занимаются специальные *почтовые серверы*. Здесь и далее под *сервером* может пониматься программное обеспечение. Таким образом, один узловой компьютер Интернета может выполнять функции нескольких серверов и обеспечивать работу различных служб.

Почтовые серверы получают сообщения от клиентов и пересылают их по цепочке к почтовым серверам адресатов, где эти сообщения накапливаются. При установлении соединения между адресатом и его почтовым сервером происходит автоматическая передача поступивших сообщений на компьютер адресата.

Адрес электронной почты состоит из двух частей, разделенных символом @. Первая часть адреса указывает конкретного пользователя. Вторая часть адреса – это доменный адрес – две последние части обозначения компьютера в адресе URL (см. ниже). Например, [ivanov@pstu.ru](mailto:ivanov@pstu.ru).

Обычно пользователи создают почтовые ящики у того провайдера, через которого они подключаются к Интернету. Существуют серверы, где можно бесплатно открыть почтовый ящик, например, [www.hotmail.com](http://www.hotmail.com).

Почтовая служба основана на двух прикладных протоколах: SMTP и POP3. По первому происходит отправка корреспонденции с компьютера на сервер, а по второму – прием поступивших сообщений. Существует большое разнообразие клиентских почтовых программ. К ним относится, например,

программа Microsoft Outlook Express (рис. 2), входящая в состав операционной системы Windows как стандартное приложение. Из специализированных почтовых программ – это программа Mozilla Thunderbird.

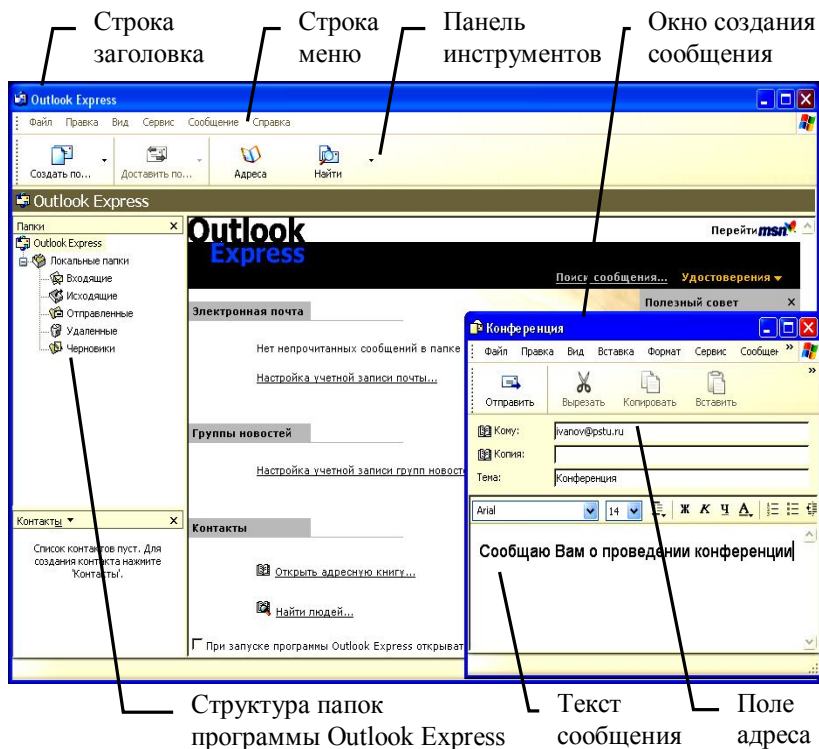


Рис. 2. Создание сообщения для отправки по электронной почте в программе Outlook Express

Так как сообщения электронной почты поступают и отправляются через почтовый сервер, программе – почтовому клиенту – требуется указать информацию об используемом сервере. Эта информация хранится в виде *учетной записи*.

В программе Outlook Express учетную запись создают командой **Сервис → Учетные записи**. В диалоговом окне **Учет-**

**ные записи в Интернете** надо щелкнуть на кнопке <Добавить> и выбрать в открывшемся меню службу, для которой создается учетная запись. Последующая информация вводится под управлением мастера и включает имя, указываемое как имя отправителя, адрес электронной почты и имя используемого сервера.

Для создания сообщения электронной почты следует щелкнуть на кнопке <Создать сообщение> на панели инструментов (см. рис. 2). При этом открывается окно **Создать сообщение**, рабочая область которого разбивается на две основные части. В верхней части располагаются поля для ввода служебной информации – адреса получателя, темы сообщения. В нижней части окна – собственно текст сообщения. В ходе создания и редактирования сообщения наличия связи с почтовым сервером не требуется.

Получение и доставка почты осуществляются по щелчку на кнопке <Отправить> в окне создания сообщения или по щелчку на кнопке <Доставить почту> в основном окне программы Outlook Express.

Сообщения электронной почты размещаются в системе «внутренних» папок программы Outlook Express. Поступившие сообщения заносятся в папку **Входящие**.

Для хранения адресов электронной почты используется специальная программа *Адресная книга*.

**Списки рассылки (Mail List)**. Достаточно большой поток почтовой информации в свой адрес можно обеспечить, подписавшись на *списки рассылки*. Это специальные тематические серверы, собирающие информацию по определенным темам и переправляющие ее подписчикам в виде сообщений электронной почты.

**Служба телеконференций (Usenet)**. Служба телеконференций похожа на циркулярную рассылку электронной почты, в ходе которой одно сообщение отправляется не одному корреспонденту, а большой группе (такие группы называются *телеконференциями* или *группами новостей*).

Вся система телеконференций разбита на тематические группы (в мире их порядка 50 000). Основной прием использования групп новостей состоит в том, чтобы задать вопрос, об-

ращаясь ко всему миру, и получить ответ или совет от тех, кто с этим вопросом уже разобрался.

Для работы со службой телеконференций существуют специальные клиентские программы. Приложение Microsoft Outlook Express, указанное выше как почтовый клиент, позволяет работать также и со службой телеконференций. Для начала работы надо настроить программу на взаимодействие с сервером групп новостей, оформить «подписку» на определенные группы и периодически получать все сообщения, проходящие по теме этой группы.

**Служба World Wide Web (WWW).** Это самая популярная служба современного Интернета. World Wide Web – это единое информационное пространство, состоящее из сотен миллионов взаимосвязанных электронных документов, хранящихся на Web-серверах. Отдельные документы, составляющие *пространство Web*, называются *Web-страницами*. Группы тематически объединенных Web-страниц называются *Web-сайтами* или просто *сайтами*.

От обычных текстовых документов Web-страницы отличаются тем, что они оформлены без привязки к конкретному носителю. Оформление выполняется непосредственно во время их воспроизведения на компьютере клиента и происходит оно в соответствии с настройками программы, выполняющей просмотр. Программы для просмотра Web-страниц называют *браузерами*. Наиболее важной чертой Web-страниц являются *гипертекстовые ссылки*. В этом случае при щелчке левой кнопкой мыши на тексте или рисунке, являющемся гиперссылкой, отправляется запрос на доставку нового документа.

Со стороны Интернета работу службы World Wide Web обеспечивают серверные и программные средства – *Web-серверы*. Со стороны пользователя работа обеспечивается клиентскими программами – *Web-браузерами* (например, Microsoft Internet Explorer). Для запуска этой программы можно использовать Главное меню (**Пуск → Все программы → Internet Explorer**). В открывшемся окне программы в поле **Адрес** необходимо ввести URL-адрес ресурса (если он известен) или восполь-

зоваться одной из поисковых систем – [www.yandex.ru](http://www.yandex.ru), [www.rambler.ru](http://www.rambler.ru), [www.google.ru](http://www.google.ru) или др.

Адрес любого файла во всемирном масштабе определяется *унифицированным указателем ресурса – URL*. Адрес URL состоит из трех частей:

1. Указание службы, которая осуществляет доступ к данному ресурсу (обычно обозначается именем прикладного протокола, соответствующего данной службе). Так, например, для службы WWW прикладным является протокол *HTTP (Hyper Text Transfer Protocol – протокол передачи гипертекста)*. После имени протокола ставится двоеточие и два знака «/»:

**http://...**

2. Указание *доменного имени* компьютера (сервера), на котором хранится данный ресурс:

**http://www.google.com...**

3. Указание полного пути доступа к файлу на данном компьютере. В качестве разделителя используется символ «/»:

**http://www.google.com/intl/ru/options**

При записи URL-адреса важно точно соблюдать регистр символов, так как в Интернете строчные и прописные символы считаются разными.

**Служба имен доменов (DNS).** В Интернете есть две разные формы записи адреса одного и того же сетевого компьютера: числовая (195.38.46.11) и доменная ([www.google.com](http://www.google.com)). Человеку неудобно работать с числовым представлением IP-адреса, зато доменное имя запоминается легко, особенно если учесть, что, как правило, это имя имеет содержание. С другой стороны, автоматическая работа серверов сети организована с использованием четырехзначного числового адреса. Поэтому необходим перевод доменных имен в связанные с ними IP-адреса. Этим и занимаются серверы службы имен доменов DNS.

**Служба передачи файлов (FTP).** Необходимость в передаче файлов возникает, например, при приеме файлов программ, при пересылке крупных документов (например, книг), а также при передаче архивных файлов, в которых запакованы большие объемы информации.



Со стороны клиента для работы с серверами FTP может быть установлено специальное программное обеспечение, хотя в большинстве случаев браузеры WWW обладают встроенными возможностями для работы и по протоколу FTP.

**IRC.** Служба *IRC (Internet Relay Chat)* предназначена для прямого общения нескольких человек в режиме реального времени. Иногда службу IRC называют *чат-конференцией* или просто *чатом*.

Одна из наиболее популярных клиентских программ для работы с серверами и сетями, поддерживающими сервис IRC – программа mIRC.exe.

**ICQ.** Эта служба предназначена для поиска сетевого IP-адреса человека, подключенного в данный момент к Интернету. Необходимость в подобной услуге связана с тем, что большинство пользователей не имеют постоянного IP-адреса. Для пользования этой службой надо зарегистрироваться на ее центральном сервере (<http://www.icq.com>) и получить персональный идентификационный номер *UIN (Universal Internet Number)*. Данный номер нужно сообщить партнерам по контактам.

При каждом подключении к Интернету программа ICQ, установленная на нашем компьютере, определяет текущий IP-адрес и сообщает его центральной службе, которая, в свою очередь, оповещает наших партнеров по контактам.

## **6.4. Вопросы компьютерной безопасности**

**Компьютерные вирусы.** Компьютерный вирус – это программный код, встроенный в другую программу или в документ, или в определенные области носителя данных и предназначенный для выполнения несанкционированных действий на компьютере. Сейчас почти во всех вирусах заложен алгоритм размножения по Сети (по электронной почте, через WEB и т.п.).

Вирусы принято делить на классы по следующим основным признакам:

- среда обитания;
- объем причиненного вреда;
- операционная система;

– алгоритм работы.

По среде обитания вирусы можно разделить:

- на файловые;
- сетевые;
- макровирусы;
- загрузочные.

По объему причиненного вреда вирусы подразделяются:

- на безвредные, т.е. никак не влияющие на работу компьютера;
- неопасные, т.е. те, которые просто себя распространяют, при этом, например, выдвигая CD-ROM или мигая лампочками на клавиатуре;
- опасные, которые могут привести к серьезным сбоям в работе компьютера;
- очень опасные, которые могут привести к потере программ, уничтожить данные, стереть необходимую для работы компьютера информацию, даже ту, которая находится в системной области данных.

Файловые вирусы – это те, которые при своем размножении используют файловую систему определенной операционной системы.

*Сетевыми* называются такие вирусы, которые при своем распространении используют возможности Интернета и локальных сетей.

Основным принципом работы сетевого вируса является возможность самостоятельно передать свой код на удаленный сервер и заставить его выполняться. Сетевые вирусы нередко называют сетевыми червями. Для своего распространения они используют ошибки и недокументированные функции сетей или операционных систем, при этом распространяясь по серверам и запуская свой код на каждом из них.

*Макровирусы* – это вирусы на макроязыках различных приложений (MS Excel, MS Word и т.п.). Для своего размножения они используют возможности макроязыков и с их помощью переносят себя из одного зараженного файла (документа или таблицы) в другие.

*Загрузочные вирусы* заражают загрузочный (boot) сектор флоппи-диска или винчестера. При заражении дисков загрузочные вирусы подставляют свой код вместо какой-либо программы, получающей управление при загрузке системы. Вирус заставляет систему при перезапуске считать в память и отдать управление не оригинальному коду загрузчика, а коду вируса.

**Средства антивирусной защиты.** Основным средством защиты информации является резервное копирование наиболее ценных данных. Резервные копии должны храниться отдельно от компьютера на внешних носителях.

Вспомогательным средством защиты информации являются антивирусные программы, которые следует регулярно обновлять, так как антивирусная программа ищет вирус путем сравнения кода программ с кодами известных ей вирусов, хранящихся в базе данных. Если база данных устарела, а вирус является новым, сканирующая программа его не обнаружит.

**Защита информации в Интернете. Понятие о шифровании информации.** Принцип действия защиты информации в Интернете основан на том, чтобы исключить возможность подбора адекватного метода для преобразования данных в информацию. Одним из приемов такой защиты является *шифрование* данных.

Обычный подход состоит в том, что к документу применяется некий *метод шифрования (ключ)*, после чего документ становится недоступен для чтения обычными средствами. Его может прочитать только тот, кто знает ключ, т.е. может применить *адекватный метод чтения*. Если в процессе обмена информацией для шифрования и чтения пользуются одним и тем же ключом, то такой криптографический процесс является *симметричным*.

Основной недостаток симметричного процесса заключается в том, что, прежде чем начать обмен информацией, надо выполнить передачу ключа, а для этого опять нужна защищенная связь, т.е. проблема повторяется.

Поэтому в настоящее время в Интернете используют *несимметричные* криптографические системы, основанные на использовании не одного, а двух ключей. Компания для работы

с клиентами создает два ключа: один – открытый (публичный) ключ, а другой – закрытый (личный) ключ. На самом деле это как бы две «половинки» одного целого ключа, связанные друг с другом. Ключи устроены так, что сообщение, зашифрованное одной половинкой, можно расшифровать только другой половинкой. Таким образом, торговая компания широко распространяет публичный ключ и надежно сохраняет закрытый. Оба эти ключа представляют собой некую кодовую последовательность.

Защиту информации принято считать достаточной, если затраты на ее преодоление превышают ожидаемую ценность самой информации. В этом состоит *принцип достаточности защиты*, которым руководствуются при использовании несимметричных средств шифрования данных.

## 7. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ПАСКАЛЬ

### 7.1. Алгоритм и его свойства

Под алгоритмизацией понимают сведение задачи к последовательности этапов, выполняемых друг за другом, так что результаты предыдущих этапов используются при выполнении последующих. *Алгоритм* – это четкое описание последовательности действий, которые необходимо выполнить для решения задачи.

Алгоритм обладает следующими свойствами: дискретностью, определенностью, результативностью, массовостью.

*Дискретность.* Процесс преобразования исходных данных в результат осуществляется дискретно, так что значения величин в каждый последующий момент времени получаются по определенным правилам из значений величин в предшествующий момент времени.

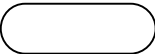
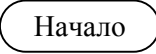
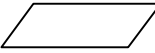

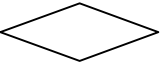
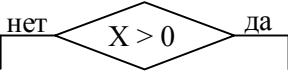
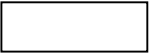
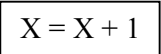

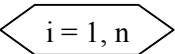
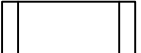

*Определенность.* Каждое правило алгоритма должно быть четким и однозначным, не допускающим двусмысленного толкования.

*Результативность.* Алгоритм должен приводить к результату за конечное число шагов.

*Массовость.* Алгоритм решения задачи разрабатывается в общем виде так, чтобы его можно было применить для класса задач, различающихся лишь исходными данными.

Программа – это окончательный вариант решения задачи на языке программирования.

Таблица 1

Символ	Описание	Пример
	Начало и окончание алгоритма	
	Ввод и вывод данных	
	Операция, определяющая выбор направления выполнения алгоритма	
	Обозначение операций присваивания	
	Обозначение заголовка цикла с параметром	
	Обозначение подпрограмм	

Алгоритм решения задачи может быть представлен графически – в виде блок-схем. Блок-схема алгоритма представляет собой совокупность блоков (табл. 1), соединенных между собой линиями связи.

## 7.2. Основные структуры алгоритмов

Теория структурного программирования доказывает, что алгоритм любой степени сложности можно построить с помощью основного базового набора структур [2].

К основным (базовым) структурам алгоритмов относятся: следование, разветвление, цикл. Каждая из структур имеет один вход и один выход.

*Следование* – это последовательное размещение блоков или групп блоков (рис. 3, а). Например, два блока S1 и S2 могут быть размещены друг за другом, при этом каждый из них в свою очередь может представлять собой любую базовую структуру.

*Разветвление* состоит из логического блока с проверкой некоторого условия P и блоков S1, S2. Разветвление может быть двух видов: полная условная конструкция (рис. 3, б) и неполная условная конструкция – обход (рис. 3, в). Полная условная конструкция применяется, когда в зависимости от условия P нужно выполнить либо S1, либо S2. Для структуры обход блок S или выполняется, или не выполняется.

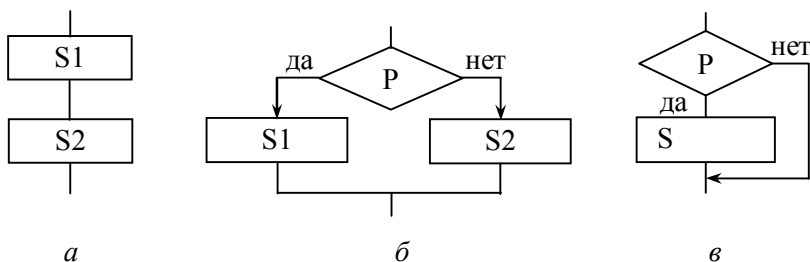


Рис. 3. Основные структуры алгоритмов: а – следование; б, в – разветвление

*Циклическими* называются алгоритмы, у которых выполнение некоторых операторов (групп операторов) осуществляется многократно с одними и теми же или модифицированными данными.

В зависимости от способа организации числа повторений различают три типа циклов: цикл с заданным условием продолжения работы (цикл-ПОКА, рис. 4, а), цикл с заданным условием окончания работы (цикл-ДО, рис. 4, б) и цикл с заданным числом повторений (цикл с параметром, рис. 4, в).

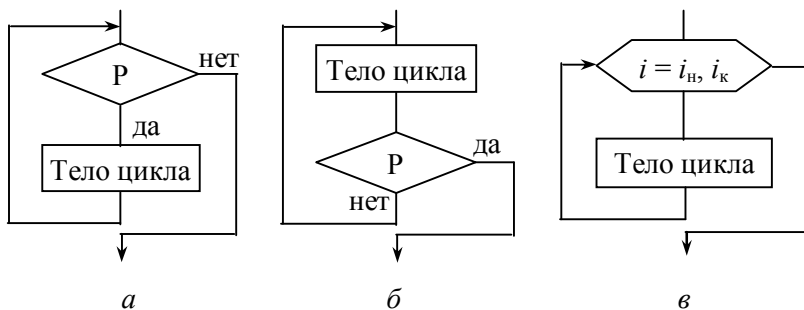


Рис. 4. Типы циклов: *а* – цикл-ПОКА; *б* – цикл-ДО; *в* – цикл с параметром

Тело цикла может включать в себя группу операторов любой степени сложности.

Для цикла-ПОКА (см. рис. 4, *а*) при выполнении условия  $P$  выполняется тело цикла; если же условие не выполняется, то работа циклической структуры заканчивается и начинает выполняться следующая структура основного алгоритма. В этом случае  $P$  – условие для продолжения цикла.

Структура цикла-ПОКА предусматривает вариант, когда тело цикла не выполняется ни разу. Такое возможно, если условие, стоящее в начале цикла, сразу же не выполняется. Когда при решении задач возникает необходимость использовать структуру, у которой тело цикла выполняется хотя бы один раз, то в этом случае применяется структура цикла-ДО (см. рис. 4, *б*). В этом случае  $P$  – условие окончания цикла, т.е. выход из цикла-ДО осуществляется при выполнении условия.

Выполнение цикла с параметром (см. рис. 4, *в*) осуществляется следующим образом: при изменении параметра  $i$  от начального значения  $i_n$  до конечного значения  $i_k$  повторяется тело цикла.

### 7.3. Алгоритмы линейной, разветвляющейся и циклической структуры

**Алгоритмы линейной структуры.** *Линейный* вычислительный процесс – это такой процесс, все вычисления которого выполняются последовательно одно за другим в естественном порядке.

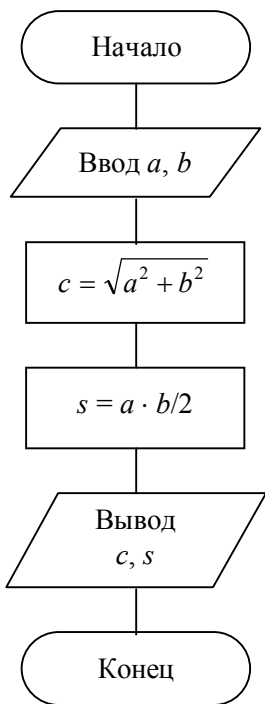


Рис. 5. Пример линейного алгоритма

*Пример.* Даны катеты  $a$ ,  $b$  прямоугольного треугольника. Найти его гипотенузу  $c$  и площадь  $s$ . Блок-схема алгоритма приведена на рис. 5.

**Алгоритмы разветвляющейся структуры.** *Разветвляющимся* (ветвящимся) называется алгоритм, который в зависимости от исходных данных или промежуточных результатов вычисления реализуется по одному из нескольких заранее предусмотренных направлений. Такие направления называются *ветвями вычислений*. Выбор той или иной ветви осуществляется в зависимости от результата проверки условия. В каждом конкретном случае алгоритм реализуется только по одной ветви, а выполнение остальных исключается.

В качестве примера использования ветвлений рассмотрим составление алгоритма для вычисления функции  $f(x)$  в зависимости от конкретных значений  $x$ ,  $a$ ,  $b$ :

$$f = \begin{cases} x + a^2, & \text{при } x \leq 1, \\ x - a \cdot b, & \text{при } 1 < x < 4, \\ x + b^2, & \text{при } x \geq 4. \end{cases}$$

Блок-схема алгоритма решения этой задачи приведена на рис. 6.



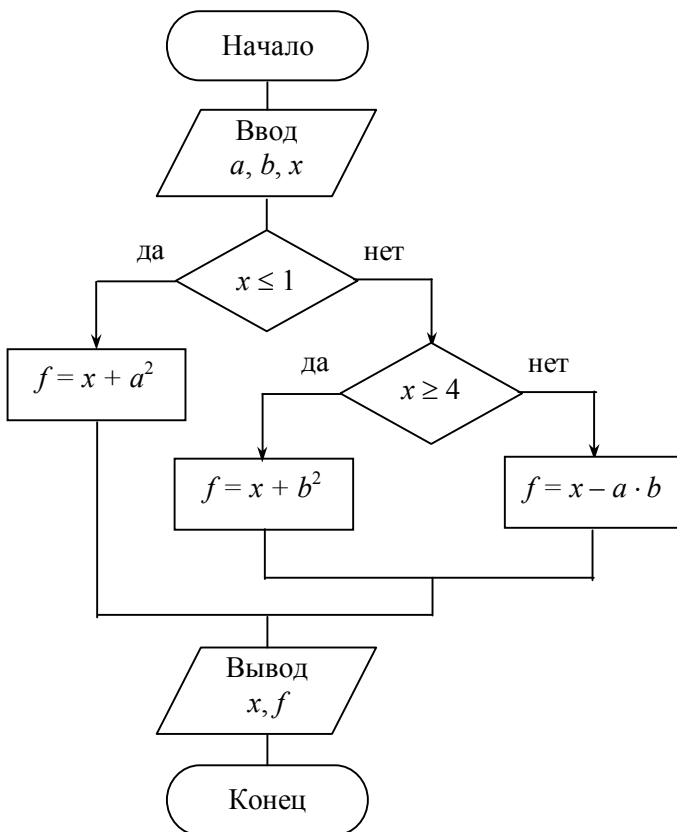


Рис. 6. Пример разветвляющегося алгоритма

**Алгоритмы циклической структуры.** Алгоритмы, в которых отдельные действия многократно повторяются, называются *циклическими*. Типовые алгоритмические структуры, реализующие циклический вычислительный процесс, приведены на рис. 4. Циклический алгоритм состоит из подготовки цикла, тела цикла и условия повторения цикла.

Характерной для этого класса вычислительных процессов является задача табулирования функции, т.е. вычисления значений функции  $y = f(x)$  на отрезке  $[x_n, x_k]$  с шагом  $\Delta x$  (рис. 7).

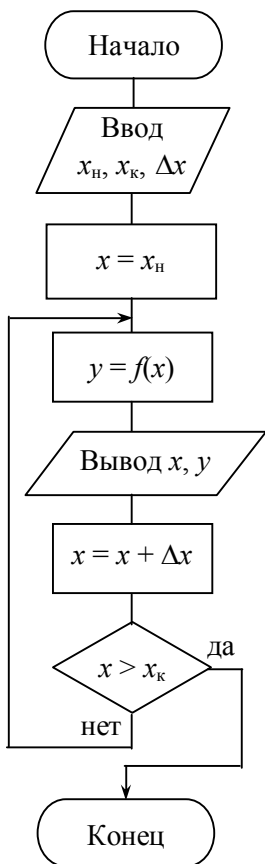


Рис. 7. Блок-схема алгоритма табулирования функции

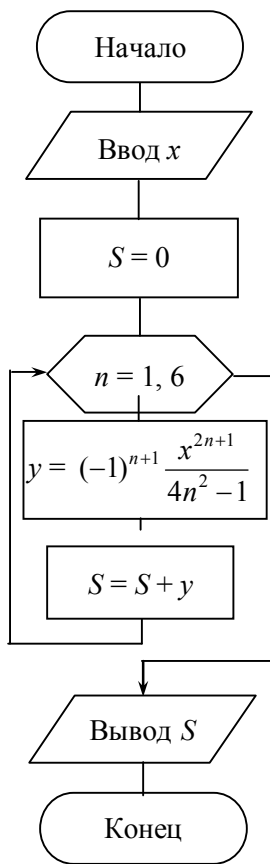


Рис. 8. Блок-схема алгоритма вычисления суммы членов ряда

Примером алгоритма циклической структуры является также задача вычисления суммы и произведения.

Если необходимо вычислить сумму значений некоторой функции  $y = f(x)$  при различных значениях аргумента, то целесообразно организовать цикл, в котором не только вычисляются текущие значения функции, но и накапливается их сумма путем прибавления полученного слагаемого к сумме предыдущих. Формула для вычисления суммы имеет следующий вид:

$$S_i = S_{i-1} + y_i.$$

При первом выполнении цикла вычисляется значение

$$S_1 = S_0 + y_1,$$

которое должно быть равно  $y_1$ . Поэтому перед циклом начальному значению суммы следует присвоить значение ноль, т.е.  $S_0 = 0$ .

Аналогично вычисляется и произведение, с той лишь разницей, что для его накопления используется формула

$$P_i = P_{i-1} \cdot y_i,$$

а начальное значение произведения должно быть равно единице, т.е.  $P_0 = 1$ .

*Пример:* вычислить  $S = \sum_{n=1}^6 (-1)^{n+1} \frac{x^{2n+1}}{4n^2 - 1}$ ,  $x = 0,3$ . Блок-

схема алгоритма решения этой задачи представлена на рис. 8.

Так как результат решения представляет собой одно число, то блок печати стоит за циклом и результаты печатаются один раз.

## 7.4. Программирование на языке Паскаль

Чтобы выполнить программу на языке Паскаль, ее необходимо ввести в память компьютера, оттранслировать, т.е. перевести на язык машинных кодов, и выполнить. Для этого на компьютере должны быть специальные средства программного обеспечения, т.е. редактор текстов, компилятор и исполнительная система. Все эти средства объединены в систему Turbo Pascal (Турбо Паскаль) [3].

**Операторы языка.** Служат для описания некоторого законченного этапа обработки данных. Операторы определяют действия над объектами программы; могут быть простыми и структурированными, т.е. состоящими из других операторов. Операторы, следующие в программе друг за другом, разделяются точкой с запятой (;).

**Идентификаторы.** Выполняют роль имен, которые даются различным программным объектам – константам, типам, пе-

ременным величинам, функциям и т.п., чтобы программисту было удобно ссылаться на эти объекты. Идентификатор может начинаться только с буквы или знака подчеркивания. Пробелы и другие специальные символы недопустимы в идентификаторах. Компилятор не делает различия между прописными и строчными буквами.

**Типы данных.** Каждая константа, переменная, элемент массива принадлежит к определенному типу данных. Тип определяет множество значений, которые может принимать объект программы, и совокупность операций, которые могут выполняться над значениями этого объекта. Тип констант распознается компилятором автоматически. Тип переменных должен быть описан в разделе описаний (в разделе переменных, начинающемся со специального слова *var*). Паскаль характеризуется разветвленной структурой типов данных (табл. 2). В табл. 2

Таблица 2

Идентификатор	Длина, байт	Диапазон (множество) значений
<i>Целые типы</i>		
integer	2	-32768...32767
byte	1	0...255
word	2	0...65535
shortint	1	-128...127
longint	4	-2147483648...2147483647
<i>Вещественные типы</i>		
real	6	$2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{38}$ (11–12)
singl	4	$1,5 \cdot 10^{-45} \dots 3,4 \cdot 10^{38}$ (7–8)
double	8	$5 \cdot 10^{-324} \dots 1,7 \cdot 10^{308}$ (15–16)
extended	10	$3,4 \cdot 10^{-4932} \dots 1,1 \cdot 10^{4932}$ (19–20)
<i>Логический тип</i>		
boolean	1	true (истина), false (ложь)
<i>Символьный тип</i>		
char	1	все символы кода ASCII

представлена информация о простых типах данных, определенных в Турбо Паскале. Для вещественных типов в скобках указано количество сохраняемых значащих цифр мантиссы в десятичном представлении числа.

**Арифметическое выражение.** Задает порядок выполнения действий над элементами данных и состоит из операндов (констант, переменных, обращений к функциям), круглых скобок и знаков операций. Выражение должно содержать данные одного типа, при этом значение выражения получается того же типа. Однако допускается использование в одном выражении данных целого и вещественного типов, результат в этом случае получается вещественного типа.

Арифметические операции приведены в табл. 3. Операции выполняются в соответствии с их приоритетом. Операции с равным приоритетом выполняются слева направо. В случае необходимости изменения порядка выполнения операции используются круглые скобки.

Таблица 3

Знак операции	Содержание	Выражение	Тип операндов	Тип результата
+	Сложение	$A + B$	Целый, вещественный	Целый, вещественный
-	Вычитание	$A - B$	Целый, вещественный	Целый, вещественный
*	Умножение	$A * B$	Целый, вещественный	Целый, вещественный
/	Деление	$A / B$	Целый, вещественный	Целый, вещественный
div	Целочисленное деление	$A \text{ div } B$	Целый	Целый
mod	Остаток	$A \text{ mod } B$	Целый	Целый

Операция *div* возвращает целую часть частного, дробная часть отбрасывается. Операция *mod* восстанавливает остаток, полученный при выполнении целочисленного деления.

**Стандартные функции.** В языке Турбо Паскаль существуют заранее разработанные подпрограммы-функции (табл. 4), которые могут использоваться в программах. Аргументы функций записываются в круглых скобках.

Таблица 4

Обозначение	Тип аргумента	Тип результата	Функция
Pi	–	real	Число $\pi = 3,1415926536$
abs(x)	integer/real	integer/real	Модуль аргумента
arctan(x)	integer/real	real	Арктангенс (радианы)
cos(x)	integer/real	real	Косинус ( $x$ в радианах)
sin(x)	integer/real	real	Синус ( $x$ в радианах)
exp(x)	integer/real	real	Экспонента $e^x$
frac(x)	integer/real	real	Дробная часть $x$
int(x)	integer/real	real	Целая часть $x$
ln(x)	integer/real	real	Натуральный логарифм
random		real	Псевдослучайное число из диапазона $[0, 1)$
random(x)	integer	integer	Псевдослучайное число из диапазона $[0, x)$
round(x)	real	integer	Округление до ближайшего целого
sqr(x)	integer/real	integer/real	Квадрат $x$
sqrt(x)	integer/real	real	Корень квадратный
trunc(x)	real	integer	Ближайшее целое, не превышающее $x$ по модулю

В Паскале нет операции или стандартной функции возведения числа в произвольную степень. Для вычисления  $a^x$  используется следующая математическая формула:  $a^x = e^{x \cdot \ln(a)}$ .

На Паскале это будет выглядеть так:  $\exp(x \cdot \ln(a))$ .

Вычисление  $\operatorname{tg}(x)$  производится с помощью выражения  $\sin(x)/\cos(x)$ .

Вычисление  $\log_b a$  производится с помощью выражения  $\ln(a)/\ln(b)$ .

Например, запишем по правилам Паскаля следующее математическое выражение:  $\frac{a + b \cdot x}{\sqrt{x^2 + 1}}$ .

На Паскале это выглядит так:  $(a + b*x)/\operatorname{sqrt}(x*x + 1)$ .

**Структура программы на Паскале.** Программа на Паскале состоит из заголовка программы и тела программы (блока), оканчивающегося точкой. В свою очередь, блок содержит *разделы описаний* и *раздел операторов* (табл. 5).

Таблица 5

Структура программы		Пример	
Заголовок программы		Program Z1;	
Оператор Uses		Uses Crt;	
Тело программы (блок)	Описание данных	1. Раздел описания меток (Label)	Label 10, 20;
		2. Раздел описания констант (Const)	Const n=10;
		3. Раздел описания типов (Type)	Type vector=array[1..n] of real;
		4. Раздел описания переменных (Var)	Var x, y: real; i, j: integer; a, b: vector;
		5. Раздел описания процедур и функций (Procedure, Function)	Procedure Tab; .....
	Описание действий	6. Раздел операторов	Begin оператор 1; оператор 2; ..... оператор N; End.

Раздел операторов является основным и присутствует в любой программе. Операторы отделяются друг от друга точкой с запятой (;). Разделы описаний (все или часть) могут отсутствовать. Разделы const, type, var могут следовать друг за другом в любом порядке и повторяться в разделе описаний несколько раз.

В { } или (\* \*) записываются комментарии к программе. Комментарий не определяет никаких действий программы и является лишь пояснительным текстом.

**Операторы ввода-вывода.** Ввод данных – это передача информации от внешних устройств в оперативную память. Вводятся, как правило, исходные данные решаемой задачи. Вывод – обратный процесс, когда данные (результаты решения задачи) передаются из оперативной памяти на внешние устройства.

**Оператор ввода** имеет следующий формат:

**read(список ввода);**

Список ввода – это последовательность имен переменных, разделенных запятыми. При выполнении этого оператора происходит остановка работы компьютера, пользователь должен ввести с клавиатуры необходимые значения переменных в том порядке, в каком они записаны в списке ввода, разделяя их пробелами. При этом вводимые значения отображаются на экране. Заканчивается ввод нажатием клавиши [Enter].

Например,

read(a,b,c);

Другой вариант оператора ввода имеет вид

**readln(список ввода);**

Этот оператор отличается тем, что после считывания последнего значения в списке ввода для одного оператора readln данные для следующего оператора ввода будут считываться с начала новой строки.

Оператор readln без параметров означает, что компьютер ожидает нажатия клавиши [Enter].

**Оператор вывода** имеет следующий формат:

**write(список вывода);**



Список вывода представляет собой выражения различных типов, разделенные запятыми.

Например,	
write(25);	выводится целая константа
write('Результаты');	выводится строковая константа
write(2*a+1);	выводится значение выражения
write('сумма=', s);	выводится список, состоящий из строковой константы и значения переменной s
write(a1, ', ', a2, ', ', a3);	выводятся значения переменных

Другая форма оператора вывода:

**writeln(список вывода);**

В этом случае после вывода всех значений из списка вывода происходит переход на новую строку. Оператор writeln без параметров означает переход на новую строку.

Рекомендуется ввод данных оформлять следующим образом:

```
write('введите коэффициенты квадратного уравнения');  
readln(a,b,c);
```

**Оператор присваивания** имеет следующий вид:

**переменная:=выражение;**

Переменная и выражение должны быть одного типа. Исключение составляет случай, когда выражение имеет целый тип, а переменная – вещественный.

*Пример.* Поменять местами значения переменных *a* и *b* (при этом используется дополнительная переменная *p*).

```
p:=a;      a:=b;      b:=p;
```

**Программирование алгоритмов разветвляющейся и циклической структуры. Условный оператор.** В Турбо Паскале управляющая структура языка, проверяющая выполнение некоторого условия и позволяющая в случае выполнения или невыполнения этого условия произвести ветвление алгоритма, реализуется с помощью условного оператора, полная форма которого следующая:

### **if *условие* then *оператор1* else *оператор2*;**

Здесь if – если, then – то, else – иначе. Операторы 1 и 2 могут быть как простыми, так и составными, представляющими собой определенную последовательность операторов. Составной оператор следует заключать в операторные скобки из слов begin и end.

Выполнение оператора: если *условие* – логическое выражение – истинно, то выполняется *оператор1*, иначе (если *условие* – логическое выражение – ложно) выполняется *оператор2*.

*Пример.*

```
If a>b then
    begin r1:=a; r2:=b end
else
    begin r1:=b; r2:=a end;
```

Краткая форма условного оператора:

### **if *условие* then *оператор*;**

В этом случае, если *условие* истинно, то выполняется *оператор*, если *условие* ложно, то управление передается следующему оператору программы.

**Операторы цикла.** В Паскале циклические алгоритмы могут быть записаны с помощью следующих операторов.

*Арифметический цикл (цикл с параметром)* с шагом +1:

### **for *параметр*:=*выражение1* to *выражение2* do *оператор*;**

Здесь for, to, do – служебные слова. *Выражение1* и *выражение2* вычисляются только один раз при входе в цикл и в процессе выполнения цикла не изменяются. В начале выполнения цикла *параметру* присваивается значение *выражения1*. Затем значение *параметра* цикла сравнивается со значением *выражения2*. Если *параметр* цикла меньше этого значения или равен ему, то выполняется тело цикла (после чего значение *параметра* изменяется на следующее значение в его типе – для целых чисел увеличивается на единицу), в противном случае выполнение цикла заканчивается.

Если тело цикла содержит более одного оператора, то оно оформляется как составной оператор с использованием операторных скобок begin...end.

*Арифметический цикл (цикл с параметром) с шагом –1:*

**for параметр:=выражение1 downto выражение2 do оператор;**

В этом случае параметр цикла изменяется по убыванию, т.е. при каждом повторении цикла параметр изменяет свое значение на предыдущее для своего типа.

*Цикл с предусловием (цикл-ПОКА):*

**while логическое выражение do оператор;**

Здесь while (пока), do (делать) – служебные слова. *Оператор* тела цикла выполняется до тех пор, пока значение *логического выражения* истинно. Для того чтобы цикл не стал бесконечным (не зациклился), необходимо наличие в теле цикла оператора, влияющего на значение *логического выражения*. Если тело цикла содержит более одного оператора, то оно оформляется как составной оператор.

*Цикл с постусловием (цикл-ДО):*

**repeat оператор until логическое выражение;**

Здесь repeat (повторять), until (до) – служебные слова. *Оператор* (тело цикла) выполняется до тех пор, пока *логическое выражение* ложно. В теле цикла можно указывать несколько операторов без использования операторных скобок. Для выхода из цикла необходимо, чтобы операторы тела цикла изменили значение *логического выражения* на истинное, иначе произойдет заикливание.

Пример использования операторов цикла при программировании циклических алгоритмов (табл. 6).

Пусть требуется вычислить сумму первых  $n$  натуральных чисел, т.е.  $S = \sum_{i=1}^n i$ . Переменная цикла  $i$  будет выполнять две функции: номер очередного слагаемого и одновременно его значение.

Таблица 6

Цикл с параметром	Цикл с предусловием	Цикл с постусловием
<pre> Program Z1;   var S, i, n: integer;       begin   writeln('задайте кол-во слагае-   мых');   readln(n);   S:=0;   for i:=1 to n do       S:=S+i;       writeln('сумма=',S);   end. </pre>	<pre> Program Z1;   var S, i, n: integer;       begin   readln(n);   S:=0;   i:=1;   while i &lt;= n do       begin       S:=S+i;       i:=i+1       end;   writeln('сумма=',S);   end. </pre>	<pre> Program Z1;   var S, i, n: integer;       begin   readln(n);   S:=0;   i:=1;   repeat       S:=S+i;       i:=i+1   until i &gt; n   writeln('сумма=',S);   end. </pre>

**Массивы.** *Массивы* представляют собой упорядоченную совокупность данных, имеющую одно имя. Каждому элементу массива соответствует выражение порядкового типа (чаще целое число), определяющее место этого элемента в массиве, которое называется *индексом*. Если для определения места элемента в массиве используется один индекс, то массив называют одномерным (вектором), два – *двумерным* (матрицей). В Паскале индекс заключается в квадратные скобки. Индекс может быть константой – V[5], переменной – V[I], выражением – V[I+3].

Массивы описываются в разделе описания переменных в следующей форме:

**var имя\_массива: array[тип\_индекса] of тип\_элементов;**

В качестве *типа индекса* чаще всего используется тип-диапазон – [N..K].

Количество элементов в диапазоне определяется следующим образом: K–N+1.

*Пример.*

```
var a: array[1..10] of integer;
```

Здесь a – имя массива, элементы которого имеют базовый тип integer, первый элемент имеет индекс 1, индекс последнего элемента 10, всего 10 элементов.

В Паскале есть возможность создавать свои типы данных, которые должны быть описаны в специальном разделе описания типов type.

*Пример.*

```
const n=10;
```

```
type vector=array[1..n] of real; {тип vector объединяет в себе  
var a, b: vector;                все одномерные массивы, со-  
                                 стоящие из n действительных  
                                 элементов}
```

Ввод и вывод массивов в Паскале осуществляется поэлементно, для чего необходимо организовать цикл.

```
for i:=1 to 10 do  
  read(a[i]);
```

```
for i:=1 to 10 do  
  write(a[i], ' ');
```

Для работы с элементами двумерного массива нужно организовать два цикла. Каждый из них отвечает за перебор значений соответствующего индекса, т.е. получается структура вложенных циклов. Параметры внешнего и внутреннего циклов разные и изменяются не одновременно: при одном значении параметра внешнего цикла параметр внутреннего цикла принимает поочередно все значения.

Двумерный массив описывается следующим образом:

```
var a: array[1..n,1..m] of integer;
```

К элементу двумерного массива обращаются следующим образом:  $a[i,j]$ , где  $i$  – номер строки,  $j$  – номер столбца. Обрабатывать элементы двумерного массива можно как по строкам, так и по столбцам:

Обработка по строкам	Обработка по столбцам
<pre>for i:=1 to n do {перебор строк}   for j:=1 to m do {перебор столбцов}     {обработка a[i,j]}</pre>	<pre>for j:=1 to m do {перебор столбцов}   for i:=1 to n do {перебор строк}     {обработка a[i,j]}</pre>

## 8. КОНТРОЛЬНЫЕ РАБОТЫ

### Общие методические указания

Контрольные работы следует выполнять по варианту, номер которого совпадает с последней цифрой номера зачетной книжки. Так, например, если номер оканчивается цифрой 8, то нужно выполнять в каждом из заданий контрольной работы задание под номером 8.

При выполнении и оформлении контрольных работ необходимо соблюдать следующие указания:

1. Контрольные работы следует выполнять на отдельных листах форматом А4.
2. На титульном листе должны быть написаны фамилия и инициалы студента, вариант и номер контрольной работы.

3. В работу должны быть включены все задания строго по варианту. Работы, содержащие не все задания, а также задания не своего варианта, не зачитываются.

### **8.1. Контрольная работа № 1**

**Цель работы:** ознакомление с операционной системой (средой) Windows; изучение структуры интерфейса пользователя; изучение способов организации и обработки объектов (папок, файлов и др.) в среде.

Ознакомление с основными принципами работы в среде текстового процессора Microsoft Word; изучение типовых приемов по формированию и обработке текстовой информации в среде.

#### **Методические указания к контрольной работе № 1**

В процессе выполнения контрольной работы необходимо ответить на вопросы заданий 1, 2 и в соответствии с пунктами задания 3 записать последовательность необходимых действий с клавиатурой и манипулятором «мышь» при обработке информации на персональном компьютере (см. приложения 1 и 2). Эта последовательность действий может быть использована при выполнении лабораторных работ.

#### **Задание 1**

1. Опишите основные функции операционной системы Windows.
2. Опишите программы, обеспечивающие взаимодействие с аппаратными средствами. Каковы особенности технологии Plug & Play?
3. Что такое оконная технология? Назовите типы окон в среде Windows и варианты их представления на экране. Что такое активное окно?
4. Что понимается под многозадачностью? Какое свойство операционной системы называется вытесняющей многозадачностью?

5. Перечислите стандартные приложения Windows и дайте их краткую характеристику.
6. Какова структура файловой системы Windows? Что называется приложением операционной системы? Где находится значок работающего приложения в Windows?
7. Перечислите объекты Windows. Какие объекты могут находиться на рабочем столе Windows? Что представляет собой Панель задач?
8. Какие типы меню используются в приложениях Windows? Перечислите пункты Главного меню (меню кнопки [Пуск]) и укажите их назначение.
9. Опишите все способы запуска приложения Windows.
10. Опишите все способы открытия созданных документов в среде Windows.

## **Задание 2**

1. Назначение и основные функции текстового процессора.
2. Опишите структуру рабочего окна текстового процессора.
3. Опишите суть операций копирования, перемещения и удаления фрагмента текста.
4. Какова роль буфера обмена (промежуточного хранения)?
5. Для каких целей производится выделение фрагментов текста?
6. Что составляет минимальный набор типовых операций текстового процессора?
7. Опишите средства автоматизации проверки правописания в документе Microsoft Word.
8. Для чего нужен режим предварительного просмотра документа и как он осуществляется?
9. Опишите режимы отображения документа в программе Microsoft Word и способы их установки.
10. Опишите специальные средства ввода и редактирования текста в документе Microsoft Word.



### Задание 3

1. Запишите последовательность действий по созданию ярлыка для приложения или документа в Windows.
2. Опишите способы переименования, копирования и перемещения объектов в Windows.
3. Как просмотреть информацию об объекте в Windows и какие сведения предоставляются в этом случае?
4. Запишите последовательность действий по открытию ранее созданных документов в Windows (приведите несколько вариантов).
5. Запишите последовательность действий по удалению и восстановлению объектов в Windows (приведите несколько вариантов).
6. Запишите последовательность действий по созданию и сохранению документа в Microsoft Word. Опишите создание формул и рисунков в документе Microsoft Word.
7. Запишите последовательность действий по копированию и перемещению фрагментов текста в Microsoft Word.
8. Запишите последовательность действий по нумерации страниц в документе Microsoft Word. Опишите приемы форматирования текста.
9. Запишите последовательность действий по созданию и редактированию таблиц в Microsoft Word.
10. Запишите последовательность действий по созданию и редактированию маркированных и нумерованных списков в Microsoft Word.

### 8.2. Контрольная работа № 2

**Цель работы:** ознакомление с основными принципами работы в среде табличного процессора Microsoft Excel; изучение типовых приемов по формированию и обработке таблиц, построению диаграмм.

#### Методические указания к контрольной работе № 2

В процессе выполнения контрольной работы необходимо ответить на вопросы задания 1 и в соответствии с пунктами за-

дания 2 записать последовательность необходимых действий с клавиатурой и манипулятором «мышь» при обработке информации на персональном компьютере (см. приложение 3). Эта последовательность действий может быть использована при выполнении лабораторных работ.

### Задание 1

1. Назначение и возможности табличного процессора Microsoft Excel. Какие преимущества может дать обработка информации с помощью электронной таблицы по сравнению с обработкой вручную? Опишите структуру рабочего окна табличного процессора. Как называются файлы электронных таблиц Microsoft Excel.

2. Какую функцию выполняет команда **Сервис** → **Параметры** (поясните назначение вкладок диалогового окна этой команды). В каких областях деятельности человека могут использоваться электронные таблицы? Приведите примеры ошибок при обработке электронных таблиц и сообщений о них.

3. Перечислите и поясните основные типы данных, которые могут быть записаны в ячейки электронной таблицы. Как осуществляется ввод данных в таблицу и в случае необходимости их корректировка? Приведите примеры использования имен ячеек. Какие действия могут быть выполнены применительно к ячейке и к диапазону ячеек?

4. Что такое формула в электронной таблице? Приведите примеры. Как сделать так, чтобы в ячейке отображался не результат вычислений по формуле, а сама формула? Опишите последовательность выполнения операций в арифметических формулах.

5. Опишите редактирование и копирование формул. Расскажите об использовании абсолютных и относительных адресов ячеек в формулах. Как это влияет на результат копирования формул?

6. Что такое функция в электронной таблице и ее типы? Приведите примеры. Назначение и использование Мастера функций. Как обозначается диапазон ячеек и несколько диапазонов ячеек в качестве аргументов каких-либо функций?

7. Перечислите типы диаграмм, используемых для интерпретации данных электронных таблиц. Опишите способы вызова на экран Мастера диаграмм.

8. Опишите возможности Microsoft Excel по редактированию диаграмм.

9. Опишите возможности табличного процессора Microsoft Excel по форматированию ячеек.

10. Расскажите о назначении двух-трех пунктов Главного меню электронной таблицы.

## Задание 2

1. Запишите последовательность действий по созданию и настройке новой рабочей книги Excel.

2. Запишите последовательность действий по формированию расчетной формулы для суммирования значений, содержащихся в любых двух интервалах ячеек.

Запишите последовательность действий, выполняемых при копировании формулы.

3. Запишите последовательность действий по выполнению типовых технологических операций с рабочими листами:

- выделить рабочий лист;
- переименовать рабочий лист;
- переместить или скопировать рабочий лист.

4. Запишите последовательность действий по созданию диаграмм.

5. Запишите последовательность действий по сохранению рабочей книги в папке Мои документы.

6. Запишите последовательность действий по выполнению настройки новой рабочей книги с использованием команды **Сервис** → **Параметры**.

7. Запишите последовательность действий по выполнению типовых технологических операций с блоками ячеек:

- форматирование блока ячеек;
- удаление блока ячеек;
- вставка блока ячеек (строк, столбцов).

8. Запишите последовательность действий по изменению размеров и места расположения диаграммы.

Запишите последовательность действий по созданию расчетной формулы с использованием стандартных функций программы Microsoft Excel.

9. Перечислите и поясните содержание основных технологических этапов работы с электронной таблицей.

10. Запишите последовательность действий по добавлению новых данных к существующей диаграмме.

### **Задание 3**

Составьте 2 примера макетов электронной таблицы, в которой присутствовала бы исходная информация и не менее трех видов производной информации (например, сумма, среднее арифметическое и т.д.). Опишите формулы, по которым получается производная информация.

### **8.3. Контрольная работа № 3**

**Цель работы:** овладение практическими навыками разработки и программирования алгоритмов решения задач.

#### **Методические указания к контрольной работе № 3**

1. Каждое задание должно включать:
  - содержание задания;
  - математическую постановку задачи, т.е. представление ее в виде уравнений, соотношений, ограничений и т.п.;
  - блок-схему алгоритма;
  - текст программы с описанием переменных, используемых в программе;
  - результаты решения задач на компьютере, сопровождая вывод наименования выводимых переменных.
2. Программу необходимо составлять в общем виде, присваивая затем переменным заданные значения. В программу должны быть включены комментарии, которые поясняют работу и назначение отдельных частей программы, характеризуют используемые переменные. Если в задании значения простых или индексированных переменных не указаны, то можно задавать любые числа.

## Задание 1

### Программирование алгоритмов разветвляющейся и циклической структуры

Реализация приема программирования – табулирования функции от одного аргумента (вычисление значений функции при изменении значения аргумента в заданном диапазоне с шагом  $\Delta x$ ).

Задание: разработать алгоритм табулирования функции. Вычислить значение функции при изменении аргумента в указанном диапазоне и с заданным шагом. Организовать вывод значения аргумента и вычисленного значения функции в виде таблицы:

ТАБЛИЦА ФУНКЦИИ  $Y(X)$

X	Y
.....	.....
.....	.....
.....	.....

Варианты заданий:

$$1. y = \begin{cases} a \cdot x^2 \cdot \ln(x) & , \text{ если } 1 \leq x \leq 2, \\ 1 & , \text{ если } x < 1, \\ e^{ax} \cdot \cos(b \cdot x) & , \text{ если } x > 2, \end{cases}$$

$$a = -0,5,$$

$$x \in [0;3],$$

$$b = 2,$$

$$\Delta x = 0,15.$$

$$2. y = \begin{cases} a \cdot x^2 + b \cdot x + c & , \text{ если } x < 1,2, \\ \frac{a}{x} + \sqrt{x^2 + 1} & , \text{ если } x = 1,2, \\ \frac{a + b \cdot x}{\sqrt{x^2 + 1}} & , \text{ если } x > 1,2, \end{cases}$$

$$a = -2,8,$$

$$x \in [1;2],$$

$$b = -0,3,$$

$$\Delta x = 0,05.$$

$$c = 4,$$

$$3. y = \begin{cases} \pi \cdot x^2 - \frac{7}{x^2} & , \text{если } x < 1,3, \\ a \cdot x^3 + 7 \cdot \sqrt{x} & , \text{если } x = 1,3, \\ \lg(x + 7 \cdot \sqrt{x}) & , \text{если } x > 1,3, \end{cases}$$

$$a = 1,5, \quad x \in [0,8;2], \\ \Delta x = 0,1.$$

$$4. y = \begin{cases} 1,5 \cdot \cos^2(x), & \text{если } x < 1, \\ 1,8 \cdot a \cdot x & , \text{если } x = 1, \\ (x - 2)^2 + 6, & \text{если } 1 < x < 2, \\ 3 \cdot \operatorname{tg}(x) & , \text{если } x \geq 2, \end{cases}$$

$$a = 2,3, \quad x \in [0,2;2,8], \\ \Delta x = 0,2.$$

$$5. y = \begin{cases} \pi \cdot x - \frac{7}{x} & , \text{если } x < 1,4, \\ a \cdot x + 7 \cdot \sqrt{x} & , \text{если } x = 1,4, \\ \ln(x + 7 \cdot \sqrt{|x + a|}) & , \text{если } x > 1,4, \end{cases}$$

$$a = 1,65, \quad x \in [0,7;2], \\ \Delta x = 0,1.$$

$$6. y = \begin{cases} x \cdot \sqrt[3]{x - a} & , \text{если } x > a, \\ x \cdot \sin(a \cdot x) & , \text{если } x = a, \\ e^{-ax} \cdot \cos(a \cdot x) & , \text{если } x < a, \end{cases}$$

$$a = 2,5, \quad x \in [1;5], \\ \Delta x = 0,5.$$

$$7. y = \begin{cases} b \cdot x - \lg(b \cdot x), & \text{если } b \cdot x < 1, \\ 1, & \text{если } b \cdot x = 1, \\ b \cdot x + \lg(b \cdot x), & \text{если } b \cdot x > 1, \end{cases}$$

$$b = 1,5, \quad x \in [0,1;1],$$

$$\Delta x = 0,1.$$

$$8. y = \begin{cases} \sqrt{a \cdot x^2 + b \cdot \sin(x) + 1}, & \text{если } x < 0,1, \\ a \cdot x + b, & \text{если } x = 0,1, \\ \sqrt{a \cdot x^2 + b \cdot \cos(x) + 1}, & \text{если } x > 0,1, \end{cases}$$

$$a = 1,5, \quad x \in [-1;1],$$

$$b = 0,4, \quad \Delta x = 0,2.$$

$$9. y = \begin{cases} a \cdot \lg(x) + \sqrt[3]{|x|}, & \text{если } x > 1, \\ 2 \cdot a \cdot \cos(x) + 3 \cdot x^2, & \text{если } x \leq 1, \end{cases}$$

$$a = 0,9, \quad x \in [0,8;2],$$

$$\Delta x = 0,1.$$

$$10. y = \begin{cases} \frac{a+b}{e^x + \cos(x)}, & \text{если } x < 2,8, \\ \frac{a+b}{x+1}, & \text{если } 2,8 \leq x < 6, \\ e^x + \sin(x), & \text{если } x \geq 6, \end{cases}$$

$$a = 2,6, \quad x \in [0;7],$$

$$b = -0,39, \quad \Delta x = 0,5.$$

## Методические указания к выполнению задания 1

Таблица функции состоит из заголовка и строк, содержащих значение аргумента в некоторой точке интервала и соответствующее значение функции. Поэтому, записывая программу, сначала программируем вывод заголовка. Затем в цикле вычисляем значение функции и выводим текущую строку таблицы.

Список переменных:

$a, b$  – коэффициенты функции;

$x_0, x_k$  – начальное и конечное значения интервала изменения аргумента;

$dx$  – шаг изменения аргумента;

$y$  – значение функции в точке  $x$ .

Текст программы, реализующей указанный алгоритм (для задания варианта 1):

```
program z1;
uses crt; {модуль расширения возможностей текстового
ввода-вывода}
var a, b, x, y, x0, xk, dx : real;
begin
  clrscr; {очистка экрана}
  write ('Введите a, b, x0, xk, dx');
  readln (a, b, x0, xk, dx);
  writeln (' ТАБЛИЦА ФУНКЦИИ Y(X) ');
  writeln ('X', ' ', 'Y(X) ');
  x := x0;
  repeat
    if (x>=1) AND (x<=2) then y := a*x*x*ln(x);
    if x<1 then y := 1;
    if x>2 then y := exp(a*x)*cos(b*x);
    writeln (x:8:3, ' ', y:8:3);
    x := x+dx;
  until x>xk;
  readln;
end.
```

С помощью процедуры writeln (x:8:3, ' ', y:8:3); осуществляется форматный вывод на экран значений  $x$  и  $y$ . Под каждое из



этих значений выделяется по восемь позиций, три из них – под дробную часть.

## Задание 2

### Программирование вычисления суммы и произведения

Методические указания к выполнению задания 2 см. на стр. 50–51.

Вычислить:

$$1. \sum_{i=1}^7 \frac{(-1)^{i+1}}{i(i+1)(i+2)}$$

$$\prod_{k=1}^7 \frac{(-1)^{k+1}}{k(k+1)}$$

$$2. \sum_{i=0}^{10} \frac{1}{4^i + 5^{i+2}}$$

$$\prod_{i=1}^{12} \frac{i^2}{i^2 + 2i + 3}$$

$$3. \sum_{i=1}^8 \frac{x + \cos(ix)}{2^i}; x = 0,6$$

$$\prod_{i=2}^6 \frac{i+1}{i+2}$$

$$4. \sum_{k=1}^6 \frac{(-1)^k}{(2k+1)k}$$

$$\prod_{k=1}^5 \left( \frac{k}{k+1} - \cos^k |x| \right); x = 0,3$$

$$5. \sum_{i=1}^{10} \frac{i+2}{i(i+1)}$$

$$\prod_{k=1}^5 \frac{k}{2^k + 3^{k+1}}$$

$$6. \sum_{n=1}^6 (-1)^{n+1} \frac{x^{2n+1}}{4n^2 + 1}; x = 0,15$$

$$\prod_{k=1}^4 \left( \frac{(-1)^k}{(2k+1)k} + 1 \right)$$

$$7. \sum_{k=1}^5 \frac{k^2}{k^2 + 3k + 5}$$

$$\prod_{i=1}^4 \frac{5^i}{3^i + 2^{i+2}}$$

$$8. \sum_{n=1}^4 \frac{(x-1)^{2n+1}}{(2n+1)(x+1)^{2n+1}}; x = 3$$

$$\prod_{k=1}^8 \frac{k+3}{k+5}$$

$$9. \sum_{k=1}^6 \frac{k^k \cdot x^{2k}}{x^{k+1}}; x = 2$$

$$\prod_{i=1}^{10} \left( 2 + \frac{1}{i^2} \right)$$

$$10. \sum_{k=1}^6 \frac{(-1)^{k+1}}{k(k+1)}$$

$$\prod_{i=2}^8 \left( 1 - \frac{1}{i^2} \right)^2$$

### Задание 3

#### Программирование задач, связанных с обработкой одномерных массивов

1. Задан одномерный массив, состоящий из 5 элементов целого типа. Найти сумму положительных элементов этого массива и максимальный элемент в массиве.

2. Найти сумму элементов одномерного массива, состоящего из 10 элементов. Разделить каждый элемент исходного массива на полученное значение. Результат получить в том же массиве.

3. Вычислить сумму и разность двух заданных одномерных массивов, состоящих из 6 элементов. Результат напечатать в виде двух параллельных столбцов.

4. Задан массив, состоящий из 5 элементов. Вычислить значения функции  $y = x^2$  при значениях аргумента, заданных в массиве  $x$ , и поместить их в массив  $y$ . Напечатать массивы  $x$  и  $y$  в виде двух столбцов.

5. Найти среднее арифметическое значение элементов массива, состоящего из 6 элементов. Преобразовать исходный массив, вычитая из каждого элемента среднее значение. Исходный и полученный массивы напечатать в виде двух параллельных столбцов.

6. Вычислить количество положительных элементов массива, состоящего из 8 элементов, и заменить их нулями (предполагается наличие и отрицательных элементов в массиве).

7. Вычислить минимальный элемент массива, состоящего из 8 элементов, и его номер. Преобразовать исходный массив, вычитая из каждого элемента минимальное значение.

8. Найти разность между максимальным элементом и первым элементом массива, состоящего из 6 элементов.

9. Найти сумму максимального и минимального элементов массива, состоящего из 5 элементов.

10. Найти среднее арифметическое положительных элементов массива, состоящего из 6 элементов (предполагается наличие и отрицательных элементов в массиве).

### Методические указания к выполнению задания 3

Так как ввод-вывод элементов массива и их обработка основаны на последовательном переборе всех его элементов, количество которых известно заранее, то при программировании для организации перебора используется оператор цикла с параметром.

В качестве примера рассмотрим задание варианта 1: задан одномерный массив, состоящий из 5 элементов целого типа. Найти сумму положительных элементов этого массива и максимальный элемент в массиве.

Текст программы:

```
program z3;
uses crt;
const n=5;
var a: array[1..n] of integer;
    s, i, max: integer;
begin
  clrscr;
  s := 0;
  max := -32768; {минимально возможное число из диапа-
                зона типа integer}
  writeln ('введите с клавиатуры элементы исходного
          массива');
  for i:=1 to n do
    begin
      readln (a[i]);
      if a[i]>0 then s:=s+a[i];
      if a[i]>max then max:=a[i];
    end;
  writeln ('сумма положительных элементов массива
          равна', s : 3);
  writeln ('максимальный элемент массива равен', max : 3);
  readln
end.
```

## ВОПРОСЫ ПО ДИСЦИПЛИНЕ «ИНФОРМАТИКА»

1. Информатика – предмет и задачи.
2. Информация. Свойства информации.
3. Носители данных. Операции с данными.
4. Кодирование текстовых данных.
5. Кодирование графических данных.
6. Кодирование звуковой информации.
7. Принцип действия компьютера.
8. Базовая аппаратная конфигурация компьютера.
9. Внутренние устройства системного блока.
10. Системы, расположенные на материнской плате.
11. Оперативная память.
12. Процессор. Основные параметры процессоров.
13. Видеокарта. Звуковая карта.
14. Периферийные устройства персонального компьютера.
15. Структура программного обеспечения компьютера.
16. Функции операционных систем персональных компьютеров.
17. Классификация прикладных программных средств.
18. Классификация служебных программ.
19. Текстовые процессоры и методы работы с ними.
20. Электронные таблицы и методы работы с ними.
21. Системы управления базами данных.
22. Компьютерные сети. Основные понятия.
23. Компьютерные сети. Уровни модели связи.
24. Компьютерные сети. Протоколы.
25. Компьютерные сети. Службы глобальной сети Интернет.
26. Вопросы компьютерной безопасности.
27. Операторы ввода/вывода. Основные правила их использования.
28. Программирование алгоритмов разветвляющейся структуры.
29. Программирование алгоритмов циклической структуры.
30. Массивы. Ввод/вывод и обработка элементов одномерного и двумерного массивов.

## ОПЕРАЦИОННАЯ СИСТЕМА WINDOWS XP

### 1. Рабочий стол Windows XP.

*Рабочий стол* – это обобщенное название рабочей области Windows XP, границами которой являются границы экрана (рис. 1.1). На Рабочем столе отображаются объекты Windows, представленные в виде значков, и элементы управления Windows, одним из которых является *Панель задач*. С точки зрения файловой системы Рабочий стол представляет собой папку. Содержимое и оформление Рабочего стола, а также приемы управления можно настраивать с помощью меню кнопки [Пуск] на Панели задач, выбрав пункт *Панель управления*.

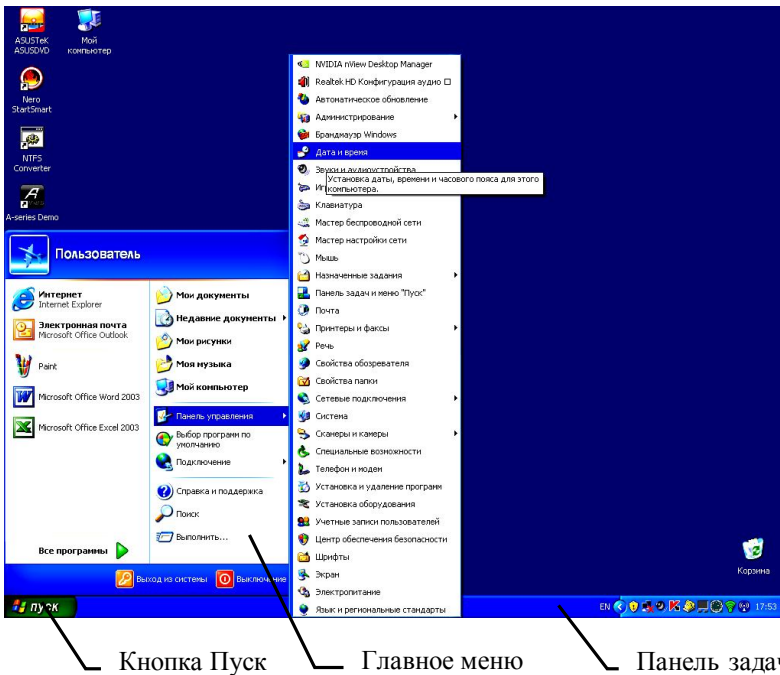


Рис. 1.1. Рабочий стол Windows XP

Значок на Рабочем столе является *графическим представлением объекта*, поэтому, например, удаление значка приводит к удалению объекта. Дополнительно существуют *ярлыки* (пиктограммы, у которых в нижнем левом углу присутствует изогнутая стрелка) – это только *указатели* на объект, следовательно, удаление ярлыка приводит к удалению указателя, но не объекта.

## 2. Окна.

Работающие приложения представлены на Рабочем столе в специальных прямоугольных областях, называемых окнами (рис. 1.2). На Рабочем столе может быть открыто одновременно несколько окон. Одно из них всегда является активным – это то окно, в котором можно в данный момент работать. Активное окно автоматически выводится на передний план, в то время как другие окна на заднем плане могут быть упорядочены произвольным образом.

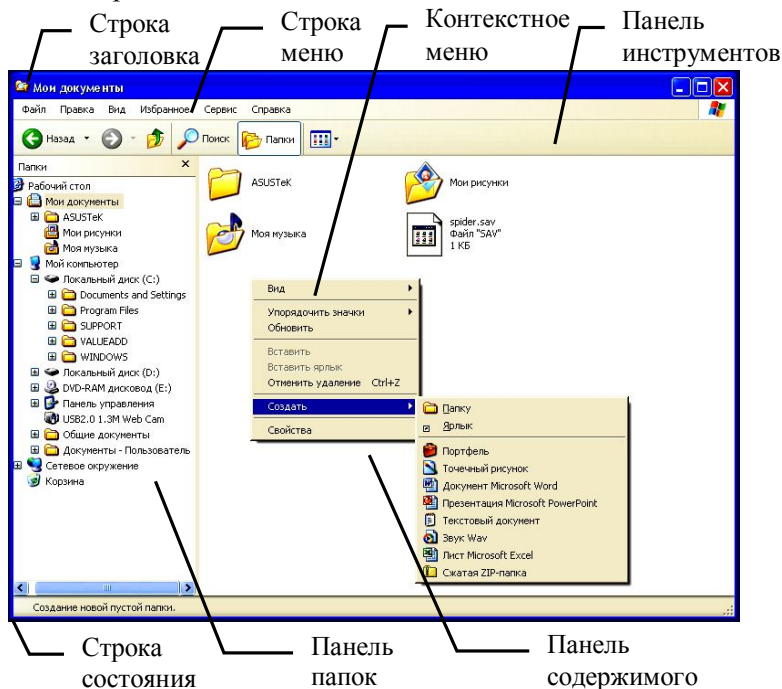


Рис. 1.2. Окно программы *Проводник*

Окна приложения и документа имеют три варианта представления на экране:

- 1) полноэкранный – окно развернуто и занимает весь экран;
- 2) нормальный – окно занимает часть экрана;
- 3) значок (пиктограмма) – окно «свернуто» в пиктограмму на Панели задач.

Для изменения варианта представления окна приложения и окна документа применяются кнопки, расположенные справа в строке заголовков окон (см. рис. 1.2).

Кроме окон приложений и документов существуют диалоговые окна, окна справочной системы и окна папок. *Диалоговые окна* – особые окна, появляющиеся на экране в случае, когда система нуждается в дополнительных указаниях пользователя или информирует его о состоянии объекта. Диалоговые окна могут содержать различные элементы управления.

### **3. Панель задач.**

Панель задач располагается в нижней части рабочего стола, содержит следующие элементы:

- Кнопка [Пуск] содержит Главное меню – один из основных системных элементов управления Windows XP (см. рис. 1.1), среди которых пункты:

- *Все программы.* Открывает доступ к иерархической структуре, содержащей указатели для запуска приложений, установленных на компьютере.

- *Недавние документы.* Открывает доступ к ярлыкам последних 15 созданных и редактируемых документов.

- *Панель управления.* Открывает доступ к основным средствам настройки операционной системы и периферийного оборудования.

- *Справка и поддержка.* Пункт входа в справочную систему Windows XP.

- *Поиск.* Открывает доступ к средствам поиска.

- *Выполнить.* Открывает окно с командной строкой для запуска приложений.

- *Выход из системы.*

- *Выключение.*

- Кнопки, соответствующие запущенным приложениям, появляются и исчезают по мере запуска и закрытия окон. Если окно свернуто, нажатием на данную кнопку осуществляется восстановление его на рабочем столе.

- Индикатор текущего алфавита, обычно русский/английский.

- Время – текущее время.

- Другие.

#### **4. Меню.**

Все необходимые для управления Windows-приложением команды собраны в меню. Выбор команды осуществляется посредством выбора соответствующего пункта меню. Некоторые пункты могут содержать вложенные меню и т.д.

Различают:

- *Главное меню – меню кнопки [Пуск]*. Содержит набор команд по управлению операционной системой Windows XP, запуску приложений, открытию и поиску документов.

- *Меню приложения*. Содержит базовый набор команд, применимых к данному приложению.

- *Контекстные меню*. Содержат ограниченный список команд, применимых к данному контексту (контекст, как правило, определяется текущей позицией курсора). Контекстное меню вызывается нажатием правой кнопки мыши.

#### **5. Запуск приложений и документов.**

Работа с документами связана с такими технологическими операциями, как запуск приложений, открытие созданных документов, их копирование и перемещение, переименование, удаление, восстановление и др. Многие из указанных операций могут быть выполнены различными способами, основанными на использовании:

- папки *Мой компьютер*;

- программы *Проводник* (см. рис. 1.2);

- команды **ФАЙЛ, Открыть** из окна приложения;

- папки *Все программы* Главного меню (меню кнопки [Пуск]);

- папки *Недавние документы* Главного меню (меню кнопки [Пуск]);



- кнопок свернутых окон *Панели задач*;
- ярлыков. Создать ярлык можно, используя контекстное меню, следующим образом (см. рис. 1.2):
  - выделить объект (например, в окне программы *Проводник*), щелкнув по нему левой кнопкой мыши;
  - вызвать контекстное меню, щелкнув правой кнопкой мыши;
  - перемещая указатель мыши по пунктам контекстного меню, выбрать пункт **Создать** → **ярлык**, щелкнув по нему левой кнопкой мыши;
  - окна <Поиск> Главного меню (меню кнопки [Пуск]).

## **6. Переименование, копирование и перемещение файлов и папок.**

Перед копированием или перемещением файла (папки) его выделяют в окне щелчком кнопки мыши по значку. Выделенный файл (папка) может быть переименован, скопирован и перемещен с помощью:

- контекстного меню;
- команд меню.

Копирование и перемещение можно также выполнить с использованием кнопок на панели инструментов.

Копирование с помощью мыши осуществляется нажатием кнопки мыши и транспортировкой значка файла (папки) в другую папку, программу, на диск или дискету, рабочий стол. Можно копировать сразу несколько файлов или папок, если их предварительно выделить.

## **7. Удаление и восстановление файлов, папок и ярлыков.**

Перед удалением или восстановлением файла (папки, ярлыка) его выделяют в окне щелчком кнопки мыши по значку.

Удаление основано на использовании:

- команды **ФАЙЛ** → **удалить**;
- контекстного меню;
- перемещения мышью значка с именем файла (папки, ярлыка) в *Корзину*.

Чтобы очистить корзину, ее следует выделить и, сделав двойной щелчок мышью, в открывшемся диалоговом окне <Корзина> выполнить команду **ФАЙЛ** → **Очистить корзину**.

Для восстановления объекта до очистки корзины следует выделить значок корзины и дважды по нему щелкнуть мышью. Откроется диалоговое окно <Корзина>. Далее следует выделить восстанавливаемый объект и выполнить команду **ФАЙЛ** → **Восстановить**.

#### **8. Просмотр информации о файле, папке или ярлыке.**

Для просмотра информации о файле, папке или ярлыке следует в контекстном меню для выбранного объекта выполнить команду **Свойства**.

## ТЕКСТОВЫЙ ПРОЦЕССОР MICROSOFT OFFICE WORD 2003

Текстовый процессор Word 2003 (рис. 2.1) позволяет создавать деловые и рекламные письма, отчеты, проекты, брошюры (т.е. практически любую печатную продукцию), объединяющие в себе данные и диаграммы из программы электронных таблиц Excel 2003, адреса и базы данных Access 2003, иллюстрации из программы подготовки материалов презентаций PowerPoint 2003.

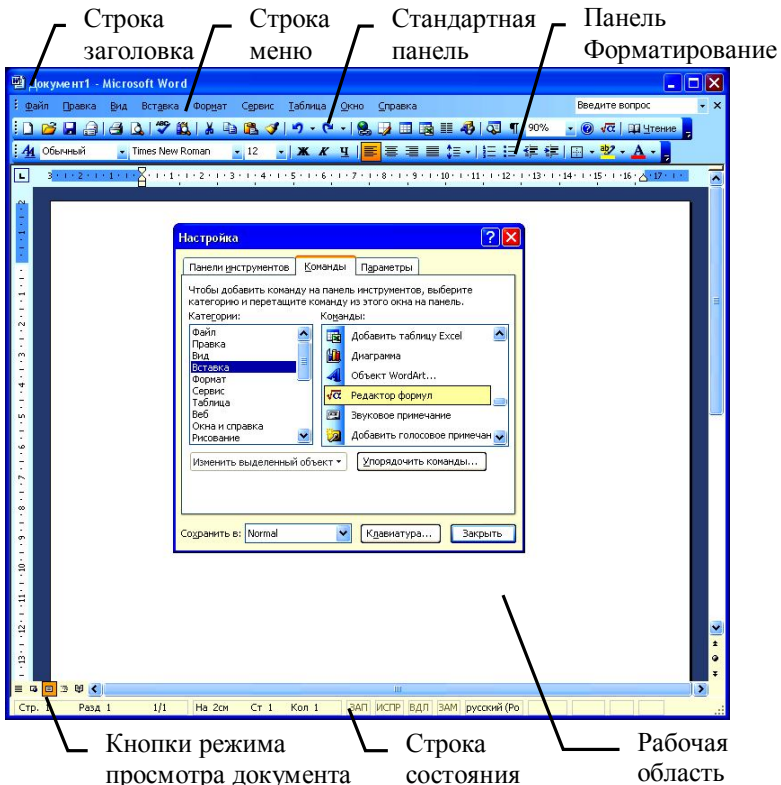


Рис. 2.1. Рабочее окно программы Word 2003

К базовым приемам работы с текстами в текстовом процессоре Microsoft Word относятся следующие:

- создание документа;
- ввод текста;
- редактирование текста;
- форматирование текста;
- сохранение документа;
- печать документа.

## 1. Структура рабочего окна программы Word 2003

(см. рис. 2.1):

– *Строка заголовка* – название программы и имя документа.

– *Строка меню* – доступ ко всем командам и средствам программы.

– *Панели инструментов*. По умолчанию доступны *Стандартная* панель и панель *Форматирование*, которые содержат кнопки для быстрого доступа к командам и инструментам Word.

– *Окно документа* (рабочая область). Служит для ввода текста и рисунков, его размер можно изменять в пределах окна Word.

– *Полосы прокрутки*. Служат для перемещения по документу.

– *Курсор* (указатель ввода) – мигающая черточка, которая показывает, где появится следующий введенный символ.

– *Кнопки режима просмотра документа*.

– *Строка состояния*.

– *Указатель мыши*. Его форма меняется в зависимости от того, на что он указывает.

Word 2003 предоставляет возможность самостоятельной настройки панелей инструментов с помощью команды **Вид** → **Панели инструментов** → **Настройка** (см. рис. 2.1). Значок выбранной команды необходимо переместить с помощью мыши в поле панели инструментов.

## 2. Создание документов.

После запуска Word автоматически создает новый документ. Дополнительно можно создать несколько новых документов с помощью команды **Файл** → **Создать**.

Создание новых документов основывается на шаблоне. *Шаблон* – это файл, содержащий свойства документа и определяющий его внешний вид, разметку страницы. Может содержать встроенный текст.

### **3. Установка параметров страницы.**

Для каждого документа можно задать отдельные параметры внешнего вида страницы (размер отступов, ориентация и другие). Для этого применяется команда **Файл** → **Параметры страницы**.

### **4. Режимы просмотра документа.**

Word позволяет отображать редактируемый документ различными способами, изменяя вид и/или объем текста и графики, отображаемых на экране. Существует четыре режима отображения. Для того чтобы изменить режим отображения документа, необходимо выбрать команду **Вид** и один из режимов:

- *обычный режим* удобен при наборе текста и редактировании. В этом режиме на экране не отображаются верхних и нижних колонтитулов, номеров страниц и т.п.;

- в *режиме Web-документа* экранное представление не совпадает с печатным. Это характерно для электронных публикаций в *World Wide Web*, поскольку заранее не известно, каким средством просмотра и на каком оборудовании будет отображаться документ;

- в *режиме разметки страницы* экранное представление полностью соответствует печатному;

- в *режиме структуры* можно отобразить только заголовки документа. Применяется для редактирования структуры документа.

Чтобы быстро изменить режим отображения, достаточно щелкнуть на одной из кнопок – <Обычный>, <Web-документ>, <Разметка страницы>, <Структура>, расположенных слева от полосы горизонтальной прокрутки в нижней части экрана (см. рис. 2.1).

### **5. Ввод текста.**

Если документ создан, можно приступить к вводу текста. По умолчанию Word автоматически переходит на следующую строку, если текст достигает правого поля. Если слово не поме-

щается в строке, то оно полностью переносится на следующую строку (если не выбрана команда **Сервис** → **Язык** → **Расстановка переносов**).

Клавишу <Enter> нужно нажимать только при переходе на новый абзац.

### **6. Редактирование текста.**

Редактирование в Word заключается в удалении, замене и перемещении отдельных слов или фрагментов текста в новое место документа.

Прежде чем удалить, переместить, скопировать или отформатировать текст, его надо выделить.

*Выделить текст* можно двумя способами: с помощью мыши и с помощью клавиатуры.

Варианты использования мыши для выделения текста приведены ниже:

Элемент текста	Действие мышью
Слово	Щелкнуть слово дважды
Рисунок	Щелкнуть один раз на рисунке
Абзац	Щелкнуть три раза любое слово абзаца
Строка	Щелкнуть слева от первого слова строки
Часть документа	Перетаскивание мыши вдоль левого поля страницы от начала фрагмента до его конца
Предложение	Удерживая нажатой клавишу <Ctrl>, щелкнуть на одном из слов предложения
Весь документ	Щелкнуть три раза на левом поле страницы

### **Копирование и перемещение текста.**

Копирование и перемещение текста состоит из двух действий: первое – указать, какой текст надо скопировать или переместить (т.е. выделить); второе – указать место, в которое нужно поместить копируемый или перемещаемый текст.

Для этого используется *Буфер обмена* – временная область памяти, где текст хранится между операциями копирования и вставки.

Для *копирования* текста надо выполнить следующие действия:

- 1) выделить текст, который необходимо скопировать;
- 2) выполнить копирование, щелкнув на кнопке <Копировать в буфер> Стандартной панели инструментов;
- 3) переместить курсор в нужное место документа;
- 4) вставить текст, щелкнув на кнопке <Вставить из буфера> Стандартной панели инструментов.

Для *перемещения* текста выполняются аналогичные действия, только вместо п. 2 необходимо щелкнуть на кнопке <Удалить в буфер> Стандартной панели инструментов.

Копирование текста методом перетаскивания и вставки (drag-and-drop) осуществляется следующим образом:

- 1) выделяется текст;
- 2) нажимается клавиша <Ctrl>;
- 3) помещается курсор мыши на выделенный абзац, нажимается и удерживается нажатой левая кнопка мыши;
- 4) перетаскивается текст в новое место;
- 5) отпускается кнопка мыши и клавиша <Ctrl>.

Перемещение текста методом drag-and-drop осуществляется аналогично, но без клавиши <Ctrl>.

### **7. Проверка орфографии.**

При вводе текста Word подчеркивает красной волнистой линией слова, которые не может найти в своем словаре. Для исправления слова нужно нажать правую кнопку мыши на нужном слове.

### **8. Предварительный просмотр документа.**

В режиме предварительного просмотра можно увидеть на экране, как будет выглядеть документ, напечатанный на бумаге. Этот режим позволяет определить, правильно ли выбраны поля и в нужном ли месте находятся номера страниц.

Для включения режима предварительного просмотра необходимо выбрать команду **Файл** → **Предварительный просмотр** или щелкнуть на кнопке <Предварительный просмотр> Стандартной панели инструментов.

## 9. Сохранение файла.

Для сохранения файла необходимо выбрать команду **Файл** → **Сохранить** или щелкнуть на кнопке <Сохранить> Стандартной панели инструментов.

Сохранить файл под другим именем можно, выбрав команду **Файл** → **Сохранить как**.

## 10. Поиск и замена текста.

Команда **Правка** → **Найти** позволяет найти нужный фрагмент текста по образцу и при необходимости заменить его на другой с помощью команды **Правка** → **Заменить**.

## 11. Печать документа.

Для печати документа необходимо выбрать команду **Файл** → **Печать** или щелкнуть на кнопке <Печать> Стандартной панели инструментов. Недостаток использования кнопки <Печать> в том, что в этом случае Word приступает к печати документа немедленно, минуя диалоговое окно печати, позволяющее указать номера печатаемых страниц, число копий и т.д.

## 12. Создание и редактирование таблиц.

Таблицы удобно использовать для организации и представления информации в виде строк и столбцов.

Для создания таблицы необходимо выбрать команду **Таблица** → **Нарисовать таблицу** или **Таблица** → **Вставить** → **Таблица**. В последнем случае в появившемся диалоговом окне задается нужное количество столбцов и строк.

Изменить количество строк или столбцов в таблице можно с помощью команд **Таблица** → **Выделить** → **Строка; Таблица** → **Удалить** → **Строка; Таблица** → **Выделить** → **Столбец** и т.д.

Средства для редактирования и форматирования таблиц находятся в меню **Таблица**.

## 13. Нумерация страниц.

Чтобы пронумеровать страницы, надо выполнить команду **Вставка** → **Номера страниц** и в появившемся диалоговом окне указать место, где необходимо расположить номера страниц.

## 14. Создание нумерованных и маркированных списков.

Для этого надо выполнить команду **Формат** → **Список** и в появившемся диалоговом окне выбрать нужный вид маркировки или нумерации.



## 15. Создание рисунков.

Непосредственно в документе можно создавать рисунки с помощью панели инструментов Рисование, появляющейся при нажатии кнопки <Панель рисования> Стандартной панели инструментов или выполнении команды **Вид** → **Панели инструментов** → **Рисование** (рис. 2.2).

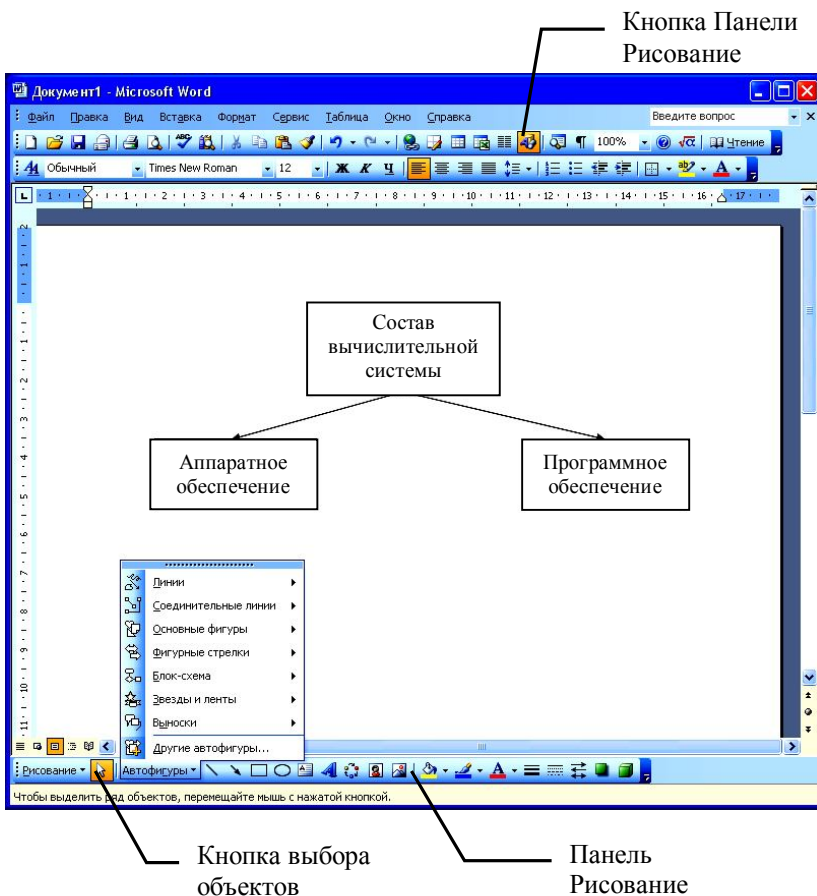


Рис. 2.2. Создание и редактирование рисунков

Основным средством панели инструментов Рисование, предназначенным для создания простейших объектов, является раскрывающийся список *Автофигуры* (см. рис. 2.2).

Создание текстовых блоков внутри элементов Автофигур выполняется с помощью команды контекстного меню **Добавить текст**.

Все свойства графических объектов можно редактировать в диалоговом окне *Формат автофигуры*, которое открывается командой **Формат** → **Автофигура** или через контекстное меню объекта.

Группировка нескольких элементов Автофигур в единое целое выполняется путем выделения их с помощью кнопки <Выбор объектов> на панели инструментов Рисование (см. рис. 2.2) и выполнения команды контекстного меню **Группировка** → **Группировать**.

#### **16. Создание и вставка формул.**

Создать в документе формулы можно с помощью команды **Вставка** → **Объект**, загрузив *Редактор формул Microsoft Equation* или щелкнув на кнопке <Добавить объект Microsoft Equation> Стандартной панели инструментов.

#### **17. Создание оглавления.**

Создать в документе оглавление можно с помощью команды **Вставка** → **Ссылка** → **Оглавление и указатели**. В появившемся окне выбрать вкладку **Оглавление** и задать необходимые параметры.

#### **18. Специальные средства ввода текста.**

*Средства отмены и возврата действий*. Все операции ввода, редактирования и форматирования текста протоколируются текстовым процессором, и потому необходимое количество последних действий можно отменить (команда **Правка** → **Отменить действие** или кнопка <Отменить действие> на панели инструментов Стандартная). Длинные последовательности действий можно отменять также с помощью списка действий (кнопка, раскрывающая список, присоединена к кнопке <Отменить действие>).

После отмены ряда действий существует возможность вернуться к состоянию, предшествовавшему отмене. Для этого

служит команда **Правка** → **Вернуть действие** или кнопка <Вернуть действие> на панели инструментов Стандартная.

*Расширенный буфер обмена* (начиная с Microsoft Word 2000) применяется для хранения фрагментов текста, например, взятых из разных источников, и последующего их использования в документе. Необходимые элементы управления находятся на панели инструментов Буфер обмена (**Вид** → **Панели инструментов** → **Буфер обмена**).

*Автотекст*. Автотекст – это режим автоматического ввода фрагментов текста. Он представлен двумя функциями: *автозавершением* и собственно *автотекстом*. Их принцип действия состоит в следующем.

Текстовый процессор хранит *словарь автотекста*, состоящий из слов и фраз, встречающихся в документах достаточно часто. При вводе первых четырех символов словарного элемента на экране появляется всплывающая подсказка с полным текстом слова или фразы. Если это то, что имел в виду пользователь, он завершает ввод всего фрагмента нажатием клавиши Enter – так работает функция *автозавершения*. Однако пользователь может самостоятельно выбрать необходимый элемент текста из списка с иерархической структурой – это функция *автотекста*. Список элементов автотекста открывается с помощью команды **Вставка** → **Автотекст** → **Автотекст**.

*Использование средства автозамены при вводе*. Это позволяет заменить ввод длинных последовательностей символов произвольным (желательно коротким) сочетанием других символов.

Настройку средства Автозамена выполняют в диалоговом окне **Сервис** → **Автозамена**. Для этого надо установить флажок **Заменять при вводе**, ввести заменяемую комбинацию в поле **Заменить**, а замещающую комбинацию в поле **На**, после чего пополнить список автозамены щелчком на кнопке **Добавить**.

*Ввод специальных и произвольных символов*. Кроме букв, символов и цифр, расположенных на клавиатуре, в текст можно вставить специальные символы в текущую позицию курсора. Эта операция выполняется командой **Вставка** → **Символ**.

## 19. Специальные средства редактирования текста.

*Использование Тезауруса.* Тезаурус представляет собой словарь смысловых синонимов, вызывающийся командой **Сервис** → **Язык** → **Тезаурус**.

*Средства автоматизации проверки правописания.* Средства автоматизации проверки правописания включают средства проверки орфографии и грамматики. Текстовый процессор позволяет реализовать два режима проверки правописания – *автоматический* и *командный*.

Для работы в автоматическом режиме надо установить флажки **Автоматически проверять орфографию** и **Автоматически проверять грамматику** на вкладке Правописание диалогового окна Параметры (**Сервис** → **Параметры** → **Правописание**). В автоматическом режиме слова, содержащие орфографические ошибки, подчеркиваются красным цветом, а выражения, содержащие грамматические ошибки, – зеленым.

В командном режиме запуск средства проверки выполняют командой **Сервис** → **Правописание**. В тех случаях, когда пользователь отказывается от предлагаемых исправлений, надо щелкнуть на кнопке <Пропустить>.

## 20. Форматирование текста.

Форматирование текста осуществляется средствами меню **Формат** или панели инструментов **Форматирование**. Основные приемы форматирования включают:

- выбор и изменение гарнитуры шрифта;
- управление размером шрифта;
- управление начертанием и цветом шрифта;
- управление методом выравнивания;
- создание маркированных и нумерованных списков;
- управление параметрами абзаца.

## 21. Приемы и средства автоматизации разработки документов.

Некоторые из приемов автоматизации ввода и редактирования текста были рассмотрены выше – это средства Автотекст, Автозамена, средства проверки правописания, средства расстановки переносов, средства поиска и замены фрагментов текста,

шаблоны документов. К ним также относятся *стили оформления абзацев*.

*Работа со стилями.* Стил ь оформления – это именованная совокупность настроек параметров шрифта, абзаца, языка и некоторых элементов оформления абзацев (линий и рамок). Настройку стиля выполняют командой **Формат** → **Стили и форматирование**. При этом можно осуществить Создание стиля (кнопка <Создать стиль>).

*Колонтитулы.* Верхний и нижний колонтитулы содержат повторяющуюся информацию вверху и внизу каждой страницы документа. Вставка верхнего или нижнего колонтитула выполняется командой **Вид** → **Колонтитулы**.

## ЭЛЕКТРОННЫЕ ТАБЛИЦЫ MICROSOFT OFFICE EXCEL 2003

Программа Microsoft Excel (рис. 3.1) предназначена для работы с таблицами данных, преимущественно числовых. При формировании таблицы выполняют ввод, редактирование и форматирование текстовых и числовых данных, а также формул.

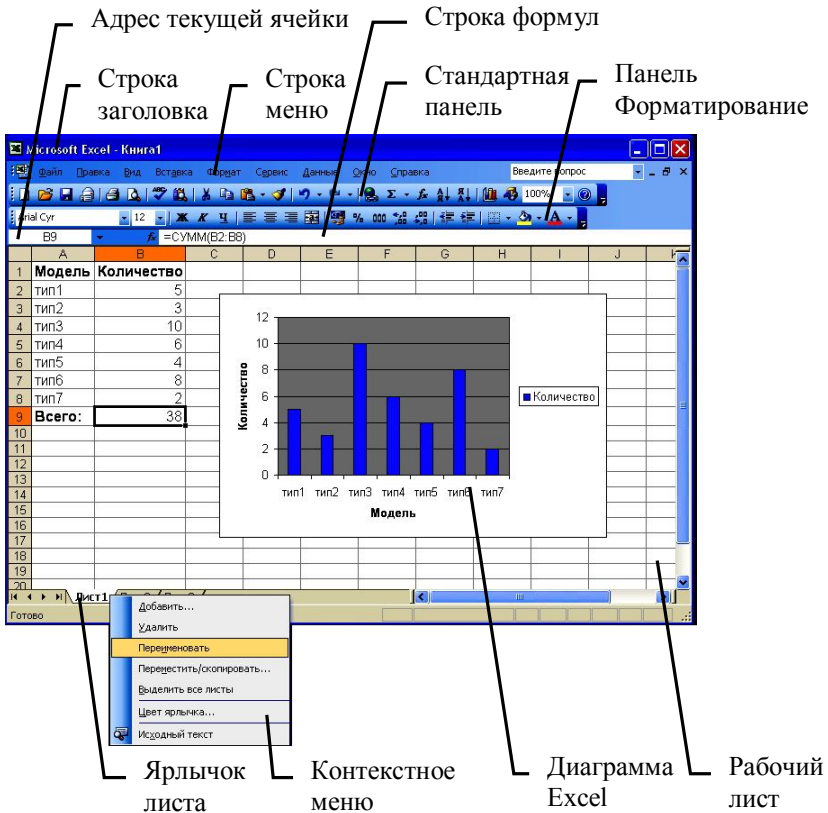


Рис. 3.1. Рабочий лист электронной таблицы Excel

## 1. Структура рабочего окна программы Microsoft Excel (см. рис. 3.1):

- *Строка заголовка* – название программы и имя документа.
- *Строка меню*. Обеспечивает доступ ко всем командам и средствам программы.
- *Панели инструментов*. По умолчанию доступны *Стандартная* и *Форматирования*; они содержат кнопки для быстрого доступа к командам и инструментам Excel.
- *Строка формул*. Отображает содержимое ячейки.
- *Окно рабочего листа* с заголовками строк (1, 2, 3 и т.д.) и заголовками столбцов (А, В, С и т.д.). Служит для ввода данных (текста, чисел, формул), построения диаграмм.
- *Полосы прокрутки*. Служат для перемещения по таблице.
- *Указатель мыши*. Форма меняется в зависимости от того, на что он указывает.

## 2. Создание новой рабочей книги и технология работы с листами.

Документ Excel называется *рабочей книгой*. Рабочая книга представляет собой набор *рабочих листов*, каждый из которых имеет табличную структуру.

Каждый рабочий лист имеет название, которое отображается на *ярлычке* листа (см. рис. 3.1). Чтобы дать листу новое название, можно использовать контекстное меню. Для этого необходимо щелкнуть на ярлычке листа правой кнопкой мыши и выбрать команду **Переименовать**. Аналогично можно выполнить вставку, удаление, перемещение и копирование рабочих листов.

Рабочий лист состоит из *строк* и *столбцов*. Столбцы обозначаются буквами (А, В, С и т.д.), а строки нумеруются (1, 2, 3 и т.д.). На пересечении столбцов и строк образуются *ячейки* таблицы. Ячейки обозначаются буквой их столбца и номером строки, в которой они расположены, например, А1.

После запуска Excel автоматически создает новую рабочую книгу. Дополнительно можно создать несколько новых рабочих книг с помощью команды **Файл** → **Создать** или кнопки < Создать > Стандартной панели инструментов. Структура

рабочей книги использует один из готовых шаблонов, который либо задается по умолчанию, либо его параметры задает пользователь.

Настройка новой книги выполняется командой **Сервис** → **Параметры**. В появившемся окне отметить флажками необходимые параметры.

### **3. Ввод, редактирование и форматирование данных.**

В ячейки рабочего листа вводятся: числа, текст, формулы и функции.

Числа, включая даты, автоматически располагаются у правого края ячейки, а текст – у левого края ячейки.

Формулы строятся как выражения для вычисления нового значения.

Функции – это встроенные в Excel формулы.

Для ввода данных в рабочую таблицу необходимо щелкнуть на ячейке, в которую хотите ввести данные. Вводимые данные появляются одновременно в ячейке и в строке формул. Заканчивается ввод данных либо нажатием клавиш со стрелками для перехода в следующую ячейку, либо нажатием клавиши <Enter> на клавиатуре.

Корректировать занесенные в таблицу данные можно двумя способами: повторно вводя новые значения в ячейки, содержащие неправильные значения, или перейдя в режим редактирования нажатием клавиши <F2>. В режиме редактирования можно исправлять данные в ячейке так, как будто вы еще не закончили ввода. Для завершения ввода нового значения в режиме редактирования нужно нажать клавишу <Enter>. В отличие от ввода в пустую ячейку, при работе в режиме редактирования нельзя закончить ввод значения нажатием клавиш управления курсором. Кроме этого, изменить содержимое ячейки можно в строке формул.

Для того чтобы полностью удалить содержимое ячейки, нужно нажать клавишу <Delete>.

Когда ячейка становится активной, ее *адрес* (буква столбца и цифра строки) появляется в поле адреса ячейки, расположенном слева в строке формул. Это поле также называется *полем имени*.



Ячейки рабочего листа имеют заданный формат, который устанавливается командой **Формат** → **Ячейки** или командой контекстного меню **Формат ячеек**.

Часто в операциях обработки данных используется не отдельная ячейка, а *диапазон ячеек*. Диапазон ячеек – прямоугольная область смежных или несмежных, расположенных в разных местах ячеек. Диапазон ячеек обозначают, указывая через двоеточие номера ячеек, расположенных в противоположных углах прямоугольника, например, **A1:C10**.

Применительно к ячейке и блоку ячеек выполняются следующие действия: форматирование (см. выше), копирование, перемещение, вставка, заполнение (команда **Правка** → **Заполнить**), удаление (команда **Правка** → **Удалить**), очистка форматов и содержимого (команда **Правка** → **Очистить**). При этом первоначально выделяется блок ячеек – объект действия, а затем выбирается команда меню для исполнения действия. Выделить прямоугольный диапазон ячеек можно протягивая указатель мыши от одной угловой ячейки до противоположной по диагонали.

#### **4. Использование в таблице формул и стандартных функций.**

Для того чтобы выполнять в Excel математические расчеты, нужно либо создать свои формулы, либо воспользоваться одной из многочисленных готовых формул, называемых функциями.

Чтобы создать простую формулу, перейдите в ту ячейку таблицы, в которой должен появиться результат вычислений, и введите адреса ячеек и знаки математических операций: + (сложение), – (вычитание), / (деление), \*(умножение), ^ (возведение в степень) или операций отношения: >, >= (не меньше), <, <= (не больше), < > (не равно).

В Excel все формулы начинаются с символа равно (=).

Предположим, что в ячейке C5 нужно получить произведение значений, содержащихся в ячейках C3 и C4. Для этого надо выполнить следующие действия:

– щелкните на ячейке C5 и нажмите клавишу со знаком равенства;

- введите C3 или щелкните на ячейке C3 (чаще значительно проще щелкнуть на нужной ячейке, чем вводить ее адрес);
- нажмите клавишу <\*> (‘звездочка’);
- введите C4 или щелкните на ячейке C4;
- нажмите клавишу <Enter> на клавиатуре.

Результат вычислений по введенной формуле появится в ячейке C5. Формула в строке формул должна выглядеть так:  $\boxed{=C3 * C4}$ .

**Копирование формул.** Процедура копирования формул очень похожа на процедуру копирования любых других данных. Эта операция может быть выполнена с помощью *маркера заполнения* – черного квадратика в правом нижнем углу рамки текущей ячейки (см. рис. 3.1). При наведении на него указатель мыши приобретает форму тонкого черного крестика. Однако при копировании формул Excel учитывает, что формулы должны быть изменены в соответствии с их новым местоположением.

При копировании формулы из ячейки C5  $\boxed{=C3 * C4}$  в ячейку D5 она примет вид:  $\boxed{=D3 * D4}$ .

Excel использует *относительную адресацию*, т.е. при копировании формулы Excel скорее копирует логику формулы, чем саму формулу. Логике, которой Excel руководствуется при копировании формулы  $\boxed{=C3 * C4}$  из ячейки C5 в ячейку D5, можно выразить так: умножить содержимое ячейки, расположенной на две строки выше, на содержимое другой ячейки, расположенной на одну строку выше.

При копировании ячеек с *абсолютной адресацией* адрес ячейки либо остается неизменным, например \$A\$4, либо корректируется частично, например \$C6 (при этом будет изменяться только номер строки) или B\$5 (в этом случае будет изменяться только обозначение столбца).

Для отображения в ячейках таблицы записанных в них формул, а не результатов расчетов по ним, необходимо выполнить команду **Сервис** → **Параметры** и на вкладке **Вид** в разделе **Параметры окна** установить флажок **Формулы**.

Excel предоставляет большое число *встроенных (стандартных) функций* для различных типов вычислений (матема-

тические, статистические, логические, текстовые, информационные и др.).

При записи формулы обращение к стандартной функции можно написать вручную или с помощью *Мастера функций* (его можно вызвать с помощью кнопки  $f_x$  на Стандартной панели инструментов) (см. рис. 3.1). Вызов функции состоит в указании в формуле *имени функции*, после которого в скобках указывается *список параметров (аргументов)*. Аргументы – это числа или адреса ячеек, которые необходимы для вычисления функции. Например, функция СУММ в качестве аргументов использует диапазоны ячеек, значения которых необходимо просуммировать, например,  $\text{=СУММ}(A1:A10)$ .

Аргументом функции СУММ может быть несколько диапазонов, которые разделяются точкой с запятой. Например,  $\text{=СУММ}(A12:A24;C18:C33)$ .

#### **5. Сообщения об ошибках.**

##### – недостаточно ширины ячейки для вывода числа.

#ДЕЛ/0! – выражение содержит деление на нуль.

#Имя? – выражение использует неопределенное значение ячейки, либо имя введено не корректно.

#Знач! – содержит недопустимый тип аргумента (например, арифметические действия выполняются над текстовыми значениями).

#ССЫЛКА! – ссылается на несуществующие ячейки или интервал ячеек, возможно, удаленные значения в ячейках.

#ПУСТО! – пытается найти несуществующее значение.

#Н/Д – ссылается на ячейку с недопустимым значением.

#ЧИСЛО! – содержит некорректную математическую операцию, например извлечение корня из отрицательного числа.

#### **6. Построение диаграмм.**

В программе Excel диаграммы графически представляют данные числового типа. Они широко используются для анализа и сравнения данных, представления их в наглядном виде.

Числовым данным рабочего листа соответствуют элементы диаграммы, которые изображаются различными геометрическими фигурами в зависимости от выбранного типа диаграммы.

При изменении исходных данных автоматически изменяется изображение элементов диаграммы по размеру или их местоположению. При изменении элемента диаграммы (увеличить или уменьшить высоту столбика, изменить местонахождение точки графика и т.п.) автоматически изменяются соответствующие числовые значения в таблицах.

Диаграмма создается с помощью *Мастера диаграмм*, вызываемого командой **Вставка** → **Диаграмма** или кнопкой *Мастер диаграмм* на Стандартной панели инструментов.

Диаграмма может располагаться на том же листе, на котором находятся данные, или на любом другом листе (часто для отображения диаграммы отводят отдельный лист).

Обычно перед вызовом *Мастера диаграмм* выделяется *диапазон ячеек – область данных*, включая названия строк и столбцов, содержащих данные для построения диаграммы (рис. 3.2). *Мастер диаграмм* осуществляет построение новой диаграммы в интерактивном режиме за четыре шага.

*Шаг 1.* Выбор типа диаграммы.

Excel 2003 позволяет строить диаграммы 14 стандартных типов. Для выбранного типа диаграммы справа указывается несколько вариантов представления данных (палитра *Вид*), из которых следует выбрать наиболее подходящий. После задания типа и вида диаграммы следует щелкнуть на кнопке <Далее>.

*Шаг 2.* Выбор данных.

Второй этап работы Мастера диаграмм служит для выбора данных, по которым будет строиться диаграмма. Если диапазон данных был выбран заранее, то в области предварительного просмотра в верхней части окна Мастера появится приблизительное отображение будущей диаграммы (значки \$ в обозначении диапазона данных указывают, что Excel использует абсолютную адресацию). Для перехода к шагу 3 (следующему диалоговому окну Мастера диаграмм) необходимо щелкнуть на кнопке <Далее>.

*Шаг 3.* Задание параметров диаграммы.

Третий этап работы Мастера диаграмм состоит в выборе оформления диаграммы (добавление легенды, ввод названия диаграммы и подписей к осям координат и т.п.). По мере того

как будут задаваться параметры диаграммы, все изменения будут отображаться на экране, показывая, как в конечном итоге будет выглядеть диаграмма.

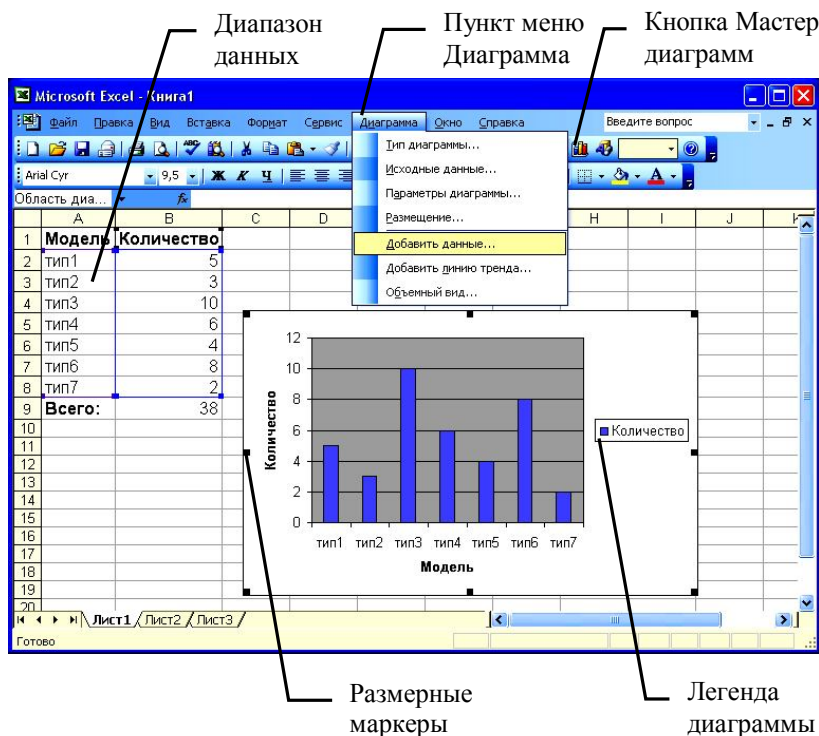


Рис. 3.2. Построение диаграмм в программе Excel

#### Шаг 4. Размещение диаграммы.

На последнем этапе работы Мастера указывается, где будет размещена диаграмма (на отдельном листе или на имеющемся). После щелчка на кнопке <Готово> диаграмма строится автоматически.

Маленькие черные квадратики в углах и посередине линий рамки, называемые размерными маркерами (см. рис. 3.2), появятся, если щелкнуть левой кнопкой мыши в поле диаграммы. Они свидетельствуют о том, что диаграмма выделена.

Чтобы изменить размер диаграммы, установите указатель мыши на один из размерных маркеров (курсор мыши изменит свое начертание на черную тонкую двустороннюю стрелку) и, удерживая нажатой левую кнопку мыши, переместите маркер в нужном направлении.

Изменить место расположения диаграммы можно, нажав левую кнопку мыши внутри диаграммы и перетащив всю диаграмму в нужное место.

## **7. Редактирование диаграмм.**

Созданные диаграммы можно корректировать вплоть до изменения состава и способа представления исходных данных, на основании которых построены диаграммы. Редактирование диаграмм выполняется как с помощью *Мастера диаграмм*, так и с помощью команд меню и панели инструментов *Диаграмма*.

Можно войти в режим редактирования диаграммы, дважды щелкнув на выбранном для изменения элементе диаграммы, которым может являться ряд данных, ось координат, заголовок диаграммы, область построения и прочее. Выйти из режима редактирования можно, щелкнув на рабочем листе вне области диаграммы.

При активизации диаграммы происходит изменение состава режимов меню программы Excel, появляются специальные режимы, содержащие команды корректировки диаграмм.

Для добавления данных рабочего листа в качестве новых рядов или элементов данных к существующей диаграмме необходимо:

- активизировать диаграмму, щелкнув по ней левой кнопкой мыши;
- выполнить команду **Диаграмма** → **Добавить данные**;
- в диалоговом окне *Новые данные* задать диапазон добавляемых данных (можно выделить соответствующий интервал данных с помощью мыши).

Форматирование элементов диаграммы выполняется с помощью пункта меню **Формат** (для выделенного элемента диаграммы) или через контекстное меню (команда **Формат**). Различные вкладки открывшегося диалогового окна позволяют изменять параметры отображения выбранного элемента диаграммы.

## СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ MICROSOFT ACCESS

Система управления базой данных (СУБД) – это комплекс программных средств, предназначенных для создания структуры новой базы, наполнения ее содержимым, редактирования содержимого, отбора отображаемых данных в соответствии с заданным критерием, их упорядочения, оформления и вывода или передачи по каналам связи.

Основным объектом базы данных являются таблицы, которые состоят из *полей* (столбцов) и *записей* (строк).

Работа с любыми объектами начинается с окна *База данных* (рис. 4.1).

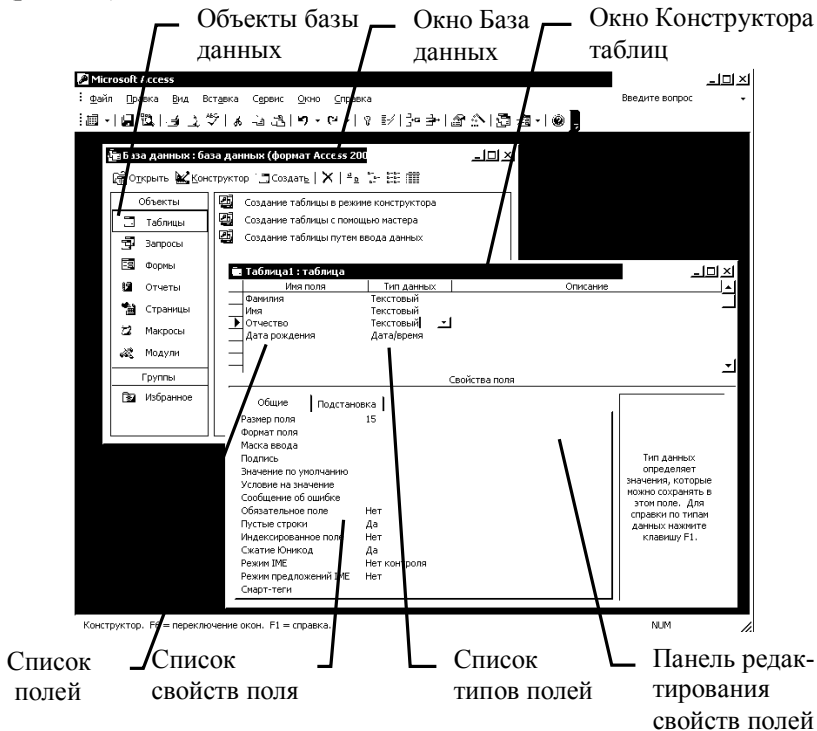


Рис. 4.1. Создание таблицы в режиме Конструктора

На левой панели данного окна сосредоточены элементы управления для вызова всех семи типов объектов программы. Создание таблиц начинается с выбора элемента управления *Таблицы*. На правой панели в этом случае представлены элементы управления для создания новой таблицы. Чтобы создать таблицу вручную, следует использовать значок *Создание таблицы в режиме конструктора*.


В первом столбце окна *Конструктора таблиц* вводят имена полей таблицы (см. рис. 4.1). Тип для каждого поля выбирают из раскрывающегося списка, открываемого кнопкой выбора типа данных. Нижняя часть окна содержит список свойств поля, выделенного в верхней части. Свойства полей не являются обязательными.

Закончив создание структуры таблицы, бланк закрывают, при этом сохраняя таблицу с каким-либо именем.

Созданную таблицу открывают в окне *База данных* двойным щелчком на ее значке. Заполнение таблицы данными производится обычным порядком (рис. 4.2).

Если возникнет необходимость изменить структуру таблицы (состав полей или их свойства), таблицу надо открыть в режиме Конструктора. Для этого ее следует выделить в окне *База данных* и щелкнуть на кнопке <Конструктор>.

При создании таблицы с помощью *Мастера таблиц* предлагается более 40 образцов таблиц, предназначенных для использования в личных или деловых целях, каждая из которых содержит соответствующий набор полей. Остается выбрать нужные поля, после чего Мастер автоматически создает таблицу.

Образцы таблиц, предоставляемых Мастером, приведены в нижней левой части окна (рис. 4.3). Для выбора поля таблицы следует выделить мышью его имя в списке образцов полей и щелкнуть кнопку , чтобы образец попал в список полей новой таблицы.

Шириной столбцов таблицы можно управлять методом перетаскивания их границ. Удобно также использовать автоматическое форматирование столбцов «по содержимому». Для этого надо установить указатель мыши на границу между столбцами (в строке заголовков столбцов), дождаться, когда указатель сменит форму, и выполнить двойной щелчок.



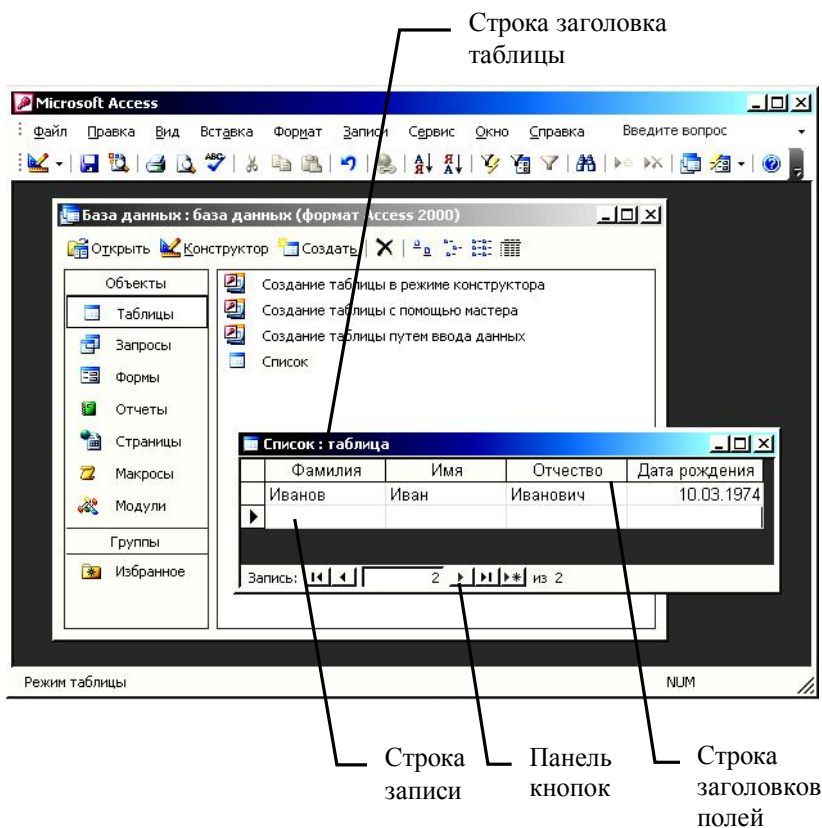


Рис. 4.2. Пример новой таблицы *Список*

После наполнения таблицы данными сохранять их не надо – все сохраняется автоматически.

*Запросы* – это объекты, которые служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде.

С помощью Access могут быть созданы следующие типы запросов:

– *запросы на выборку* (позволяют выбрать данные из полей таблиц, на основе которых сформирован запрос);

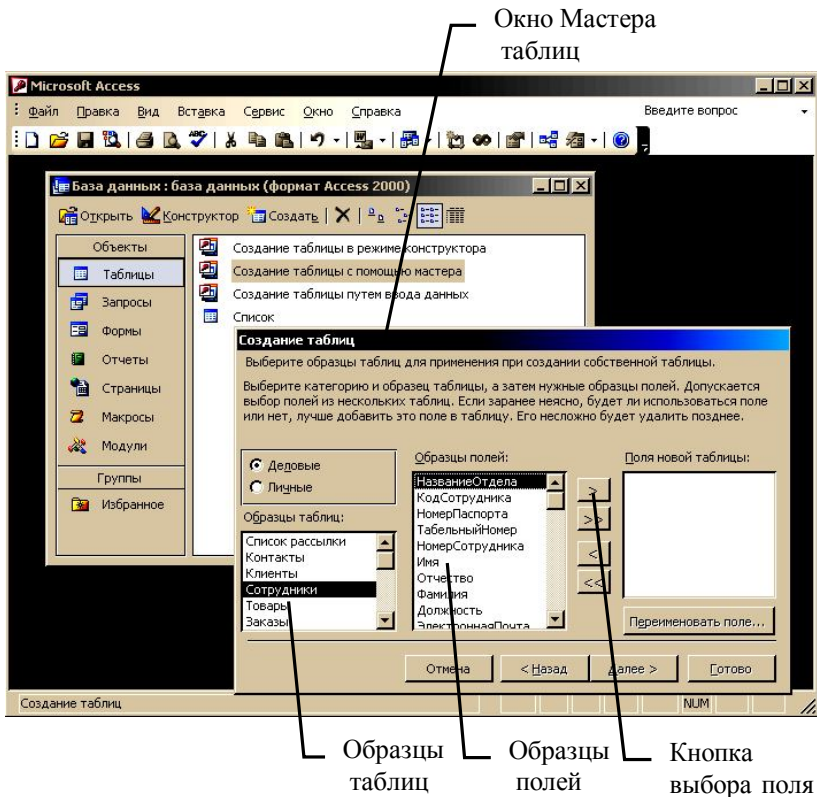


Рис. 4.3. Создание таблицы с помощью Мастера

- *запросы с параметром* (интересны тем, что критерий отбора может задать сам пользователь, введя нужный параметр при вызове запроса);
- *итоговые запросы* (производят математические вычисления по заданному полю и выдают результат);
- *запросы на изменение* (позволяют автоматизировать заполнение полей таблиц);
- *перекрестные запросы* (позволяют создавать результирующие таблицы на основе результатов расчетов, полученных при анализе группы таблиц);

– *специфические запросы SQL* (запросы к серверу базы данных, написанные на языке запросов SQL).

*Формы* – это объекты, предназначенные, в основном, для ввода данных. В форме можно разместить элементы управления, применяемые для ввода, изображения и изменения данных в полях таблицы.

*Отчеты* служат для форматированного вывода данных на печатающее устройство. Средством автоматизированного создания отчетов является *Мастер отчетов*. Он работает в шесть этапов. При этом выполняется выбор базовых таблиц и полей, отображаемых в отчете, выбор полей группировки, выбор полей и методов сортировки, выбор формы печатного макета и стиля оформления.

*Страницы* – это особый объект, выполненный в коде HTML, размещаемый на Web-странице. Сам по себе этот объект не является базой данных, но содержит компоненты, через которые осуществляется связь переданной Web-страницы с базой данных, остающейся на сервере.

*Макросы и модули* предназначены как для автоматизации повторяющихся операций при работе с СУБД, так и для создания новых функций путем программирования на языке Visual Basic for Applications.

## МАТЕМАТИЧЕСКИЙ ПАКЕТ MATHCAD

Интерфейс Mathcad аналогичен интерфейсу других Windows-приложений (рис. 5.1).

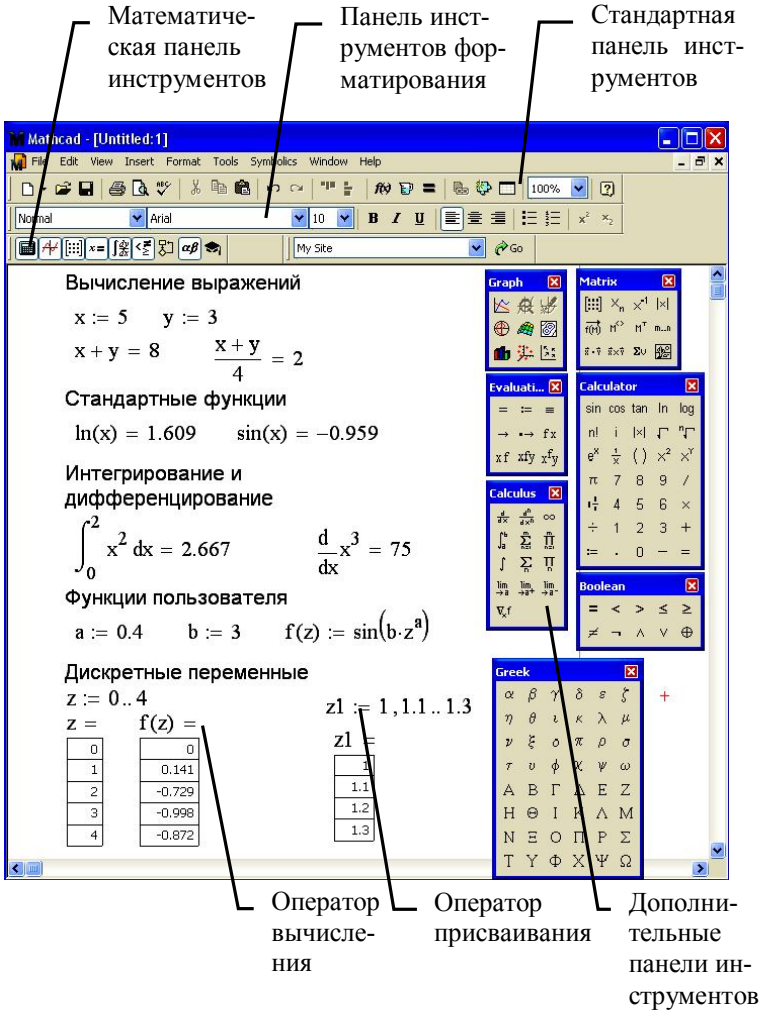


Рис. 5.1. Вычисления в программе Mathcad

Mathcad предназначен для решения следующих задач:

- вычисление результатов математических операций, в которых участвуют числовые константы, переменные и размерные физические величины;

- операции с векторами и матрицами;

- решение уравнений и систем уравнений;

- статистические расчеты и анализ данных;

- построение графиков;

- тождественные преобразования выражений, аналитическое решение уравнений и систем;

- дифференцирование и интегрирование (аналитическое и численное);

- решение дифференциальных уравнений и др.

Для ввода элементов формул предназначены дополнительные панели математической панели инструментов (см. рис. 5.1):

- Graph (График) – шаблоны графиков.

- Matrix (Матрица) – шаблоны матриц и матричных операций.

- Evaluation (Оценка) – операторы присваивания значений и вывода результатов расчета.

- Calculator (Калькулятор) – шаблоны основных математических операций.

- Calculus (Вычисления) – шаблоны дифференцирования, интегрирования, суммирования.

- Boolean (Булевы операторы) – логические (булевы) операторы.

- Greek (Греческие буквы).

- Programming (Программирование) – операторы для создания программных модулей.

- Symbolic (Символьные вычисления) – операторы символьных вычислений.

Документ программы Mathcad содержит два вида объектов: *формулы* и *текстовые блоки*. В ходе расчетов формулы обрабатываются последовательно, слева направо и сверху вниз, а текстовые блоки игнорируются.

Ввод информации осуществляется в месте расположения курсора. Программа Mathcad использует три вида курсоров. Если ни один объект не выбран, используется красный *крестообразный курсор*, определяющий место создания следующего объекта. При вводе формул используется синий *уголковый курсор*, указывающий текущий элемент выражения. Чтобы выделить элементы формулы, которые в рамках операции должны рассматриваться как единое целое, используют клавишу <Пробел>. При каждом ее нажатии уголковый курсор расширяется, охватывая элементы формулы, примыкающие к текущему элементу. При вводе данных в текстовый блок применяется *текстовый курсор* в виде вертикальной черты.

Редактирование введенных выражений производится обычным для всех Windows-приложений способом.

Mathcad воспринимает прописные и строчные буквы как разные идентификаторы.

При расчетах по формулам используются следующие операторы:

- оператор присваивания ( $:=$ ) (см. рис. 5.1);
- оператор вычисления ( $=$ ) (см. рис. 5.1);
- оператор аналитического (символьного) вычисления ( $\rightarrow$ ) (рис. 5.2);
- глобальный оператор присваивания ( $\equiv$ );
- знак логического равенства ( $=$ ).

В Mathcad можно использовать *стандартные встроенные функции* (кнопка  $f(x)$  на стандартной панели инструментов), а также *функции, определенные пользователем* (см. рис. 5.1). Функция пользователя определяется следующим образом: слева указывается название функции, а справа после оператора присваивания ( $:=$ ) – вычисляемое выражение. Переменные величины, входящие в правую часть, должны быть записаны в качестве параметров в скобках после имени функции. Все величины из правой части, не входящие в параметры левой части, должны быть заданы численно левее и выше функции пользователя.

Для получения таблицы значений функции или ее графика используются *дискретные переменные*, определяющие ряд значений. С помощью дискретной переменной можно задавать как

целые, так и дробные значения переменной, но обязательно равноотстоящие друг от друга ( $z$  и  $z1$  на рис. 5.1). Дискретная переменная – диапазон – определяется первым, вторым и последним элементами:  $x := a, a + \frac{b-a}{n} .. b$ , соответственно задает ряд чисел, где  $a$  – первое,  $a + \frac{b-a}{n}$  – второе,  $b$  – последнее число,  $n$  – число интервалов, на которые разбит отрезок от  $a$  до  $b$ . Если интервал между числами равен 1, то второй элемент отсутствует. Дискретная переменная задается с помощью кнопки  – Range Variable панели инструментов Matrix (Матрица).

Для построения двухмерного графика функции надо выполнить следующие действия (см. рис. 5.2):

1) установить крестообразный курсор в то место, где должен быть построен график;

2) на математической панели Graph (График) щелкнуть на кнопке X-Y Plot (Двухмерный график);

3) в появившемся на месте курсора шаблоне двухмерного графика ввести на оси абсцисс имя аргумента, на оси ординат – имя функции;

4) щелкнуть мышью вне шаблона графика – для заданного диапазона изменения аргумента график будет построен.

Если диапазон значений аргумента не задан, по умолчанию график строится в диапазоне значений аргумента от  $-10$  до  $10$ .

Чтобы в одном шаблоне разместить несколько графиков, следует, набрав на оси ординат имя первой функции, нажать клавишу  (запятая), и в появившемся месте ввода (черном квадратике) вписать имя второй функции и т.д. Если функции имеют разные аргументы, то на оси абсцисс надо ввести также через запятую имена аргументов функций.

Для форматирования графика необходимо дважды щелкнуть мышью в области графика – откроется диалоговое окно форматирования графика.

Определить вектор или матрицу можно следующим образом (см. рис. 5.2):

1) введите имя матрицы и оператор присваивания ( $:=$ );

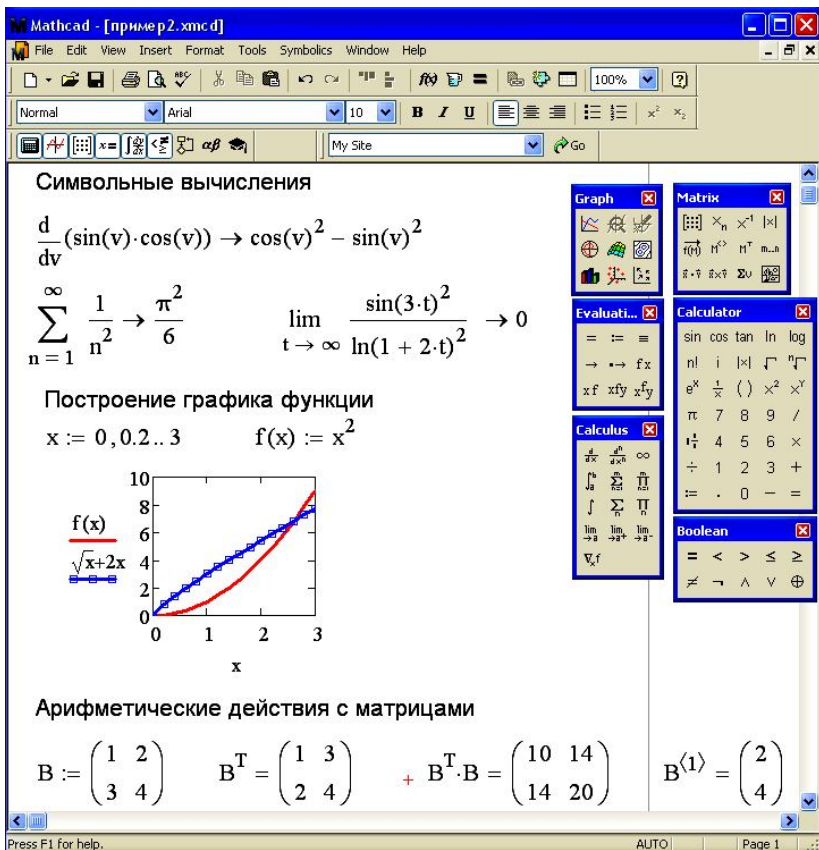


Рис. 5.2. Символьные вычисления, построение графика функции и арифметические действия с матрицами

2) на панели Matrix (Матрица) щелкните на кнопке Matrix or Vector (Матрица или вектор). Откроется диалоговое окно, в котором введите число строк и число столбцов матрицы и щелкните на кнопке ОК. На экране появится шаблон матрицы;

3) каждое место ввода в шаблоне заполните числами.

Нумерация элементов массива (вектора или матрицы) может начинаться с 0, 1 или любого другого числа. Порядком нумерации элементов массива управляет встроенная переменная



ORIGIN. По умолчанию ORIGIN=0. Это означает, что первый элемент вектора имеет номер 0. Чтобы нумерация элементов векторов и матриц начиналась с 1, надо перед вводом матрицы набрать следующую строку: ORIGIN:=1.

Доступ к любому элементу матрицы можно получить через имя матрицы с двумя индексами (например,  $B_{1,1}$ ). Первый индекс обозначает номер строки, второй – номер столбца. Произвольный элемент вектора задается одним индексом. Для набора нижнего индекса используется кнопка Subscript панели Matrix.

Чтобы из матрицы выделить вектор (один из столбцов матрицы), используется верхний индекс – номер столбца в угловых скобках, например,  $B^{<1>}$  (см. рис. 5.2). Для выполнения этой операции необходимо ввести имя матрицы и щелкнуть на кнопке  $\boxed{M^{<>}}$  – Matrix Column (Столбец матрицы) панели инструментов Matrix (Матрица).

Mathcad позволяет выполнять с матрицами основные арифметические действия, включая сложение, вычитание и умножение, а также операции транспонирования, обращения, вычисления определителя матрицы, нахождения собственных чисел и собственных векторов и т.д. (см. рис. 5.2).

Для численного поиска корней уравнений вида  $f(x) = 0$  в программе Mathcad используется функция  $root(f(x), x)$ . Перед вызовом функции  $root$  необходимо присвоить искомой переменной  $x$  начальное значение. Например,

$x:=1$

$root(2 \cdot \sin(x) - x, x) = 1.895.$

Если уравнение имеет несколько корней, то результат зависит от выбранного начального приближения.

Для решения системы уравнений (неравенств) используют блок *решения*, который начинается с ключевого слова *given* (дано) и заканчивается вызовом функции *find* (найти). Между ними располагают уравнения (неравенства), входящие в систему. При этом между левой и правой частями уравнений должен стоять знак логического равенства (=) с панели инструментов Boolean (Булевы операторы). Перед решением системы уравнений необходимо задать начальные приближения для всех неизвестных. Например,

$x:=0$        $y:=0$

*given*

$x + y = 1$

$x^2 + y^2 = 4$

$$\mathit{find}(x, y) = \begin{pmatrix} 1.823 \\ -0.823 \end{pmatrix}$$

Функция *find* возвращает вектор, содержащий вычисленные значения неизвестных.

### СПИСОК ЛИТЕРАТУРЫ

1. Информатика: базовый курс: учеб. пособие для вузов / под ред. С.В. Симоновича. – 2-е изд. – СПб.: Питер, 2008. – 639 с.
2. Аляев Ю.А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic: учеб.-справ. пособие / Ю.А. Аляев, О.А. Козлов. – М.: Финансы и статистика, 2002. – 320 с.
3. Семакин И.Г. Основы программирования: учеб. / И.Г. Семакин, А.П. Шестаков. – 5-е изд. – М.: Академия, 2006. – 431 с.
4. Информатика: учеб. для вузов / под ред. Н.В. Макаровой [и др.]. – 3-е изд. – М.: Финансы и статистика, 2007. – 765 с.
5. Макаров Е.Г. Инженерные расчеты в Mathcad: учеб. курс / Е.Г. Макаров. – СПб.: Питер, 2003. – 448 с.

Учебное издание

**ЩАПОВА Ирина Николаевна**

**ИНФОРМАТИКА**

Учебное пособие

Редактор и корректор *О.Н. Довбилкина*

---

Подписано в печать 26.08.08. Формат 60×90/16.

Усл. печ. л. 7,25.

Тираж 100 экз. Заказ № 187/2008.

---

Издательство

Пермского государственного технического университета.

Адрес: 614990, г. Пермь, Комсомольский пр., 29, к. 113.

Тел. (342) 219-80-33.