

Экзамен

без
проблем

Наглядно

доступно

И.А. Трофимова
О.В. Яровая

ИНФОРМАТИКА

в схемах
и таблицах

эффективная подготовка к ЕГЭ



наглядно /
доступно

И.А. Трофимова
О.В. Яровая

ИНФОРМАТИКА

в схемах
и таблицах



МОСКВА 2012

Т 76 **Трофимова И. А.**
Информатика в схемах и таблицах / И. А. Трофимова,
О. В. Яровая. — М. : Эксмо, 2012. — 160 с. — (Наглядно и до-
ступно).

ISBN 978-5-699-35281-4

В издании в сжатой, концентрированной форме приводится основной теоретический материал, охватывающий школьный курс информатики. Термины, определения объединены в наглядные логические модули, позволяющие лучше понять и усвоить информацию.

Пособие окажет учащимся существенную помощь в подготовке к единому государственному экзамену по информатике.

УДК 373.167:004
ББК 32.81я721

ISBN 978-5-699-35281-4

© Трофимова И. А., Яровая О. В., 2011
© Оформление. ООО «Издательство «Эксмо», 2011

СОДЕРЖАНИЕ

РАЗДЕЛ 1. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

1.1. Информация и ее кодирование	7
Информация, ее виды и свойства.	
Информационные процессы	7
Язык как способ представления и передачи информации	10
Процесс передачи информации. Источники и приемники информации. Сигнал, кодирование и декодирование. Причины искажения информации при ее передаче	12
Скорость передачи информации и пропускная способность канала связи	13
Методы измерения количества информации: алфавитный и вероятностный	13
Единицы измерения количества информации	15
Представление информации в компьютере.	
Кодирование чисел	16
Кодирование текстовой информации	17
1.2. Алгоритмизация и программирование	19
Общее понятие об алгоритме.	19
Формы представления алгоритмов.	21
Использование основных алгоритмических конструкций: следование, ветвление, цикл	25
Использование переменных. Объявление переменной (тип, имя, значение). Локальные и глобальные переменные	34
Работа с массивами	36
1.3. Моделирование и компьютерный эксперимент	38
Модели. Виды моделей. Понятие моделирования	38
Компьютерное моделирование.	40

1.4. Компьютер как средство автоматизации информационных процессов	42
Типы компьютеров, их основные характеристики и области использования	42
Архитектура современного компьютера	43
Файлы и файловые системы	47
Программное обеспечение компьютера	49
Графический интерфейс операционных систем и приложений	51
Технологии и средства защиты информации от разрушения и несанкционированного доступа	54
1.5. Основы социальной информатики	59
Основные этапы становления информационного общества	59
Информационная культура	64
Этические и правовые нормы информационной деятельности людей	64

РАЗДЕЛ 2. ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

2.1. Технология создания и обработки текстовой информации	67
Текстовые редакторы и документы	67
Создание, редактирование и форматирование текстового документа	68
2.2. Технология создания и обработки информации в электронных таблицах	81
Ввод и редактирование данных в электронных таблицах, операции с данными	81
Операции с данными	85
Типы и формат данных. Работа с формулами. Абсолютная и относительная ссылки. Использование функций	88

Визуализация данных с помощью диаграмм и графиков. Построение графиков элементарных функций	96
2.3. Технология обработки графической и звуковой информации	100
Растровая графика. Растровые графические объекты и операции с ними	100
Векторная графика. Векторные графические объекты и операции над ними	106
2.4. Создание и редактирование цифровых звукозаписей	109
Звук. Запись и воспроизведение звука	109
2.5. Компьютерные презентации: типы слайдов, мультимедиа эффекты, организация переходов между слайдами	112
Понятие компьютерной презентации	112
Основные типы слайдов	112
Порядок создания презентации в программе PowerPoint из пакета Microsoft Office 2007	115
2.6. Технология хранения, поиска и сортировки информации в базах данных	117
Структура базы данных (поля и записи)	117
Системы управления базами данных (СУБД)	120
Инструменты СУБД Access для создания таблиц	122
Инструменты СУБД Access для управления видом представленных данных	123
Инструменты СУБД Access для обработки данных	124
Инструменты СУБД Access для вывода данных	129
2.7. Средства и технологии обмена информацией с помощью компьютерных сетей	129
Базовые принципы организации и функционирования компьютерных сетей	129
Локальные и глобальные вычислительные сети	131
Адресация в сети Интернет	134

2.8. Услуги компьютерных сетей: электронная почта, телеконференция, World Wide Web	135
Электронная почта	135
Телеконференция (группа новостей)	137
Всемирная паутина (World Wide Web)	138
2.9. Поиск информации в Интернете	140
2.10. Технологии программирования	143
Чтение короткой (30–50 строк) простой программы на алгоритмическом языке (языке программирования)	143
Поиск и исправление ошибок в небольшом фрагменте программы (10–20 строк)	149
Создание собственной программы (30–50 строк) для решения простых задач	153

РАЗДЕЛ 1. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

1.1. ИНФОРМАЦИЯ И ЕЕ КОДИРОВАНИЕ

Информация, ее виды и свойства. Информационные процессы

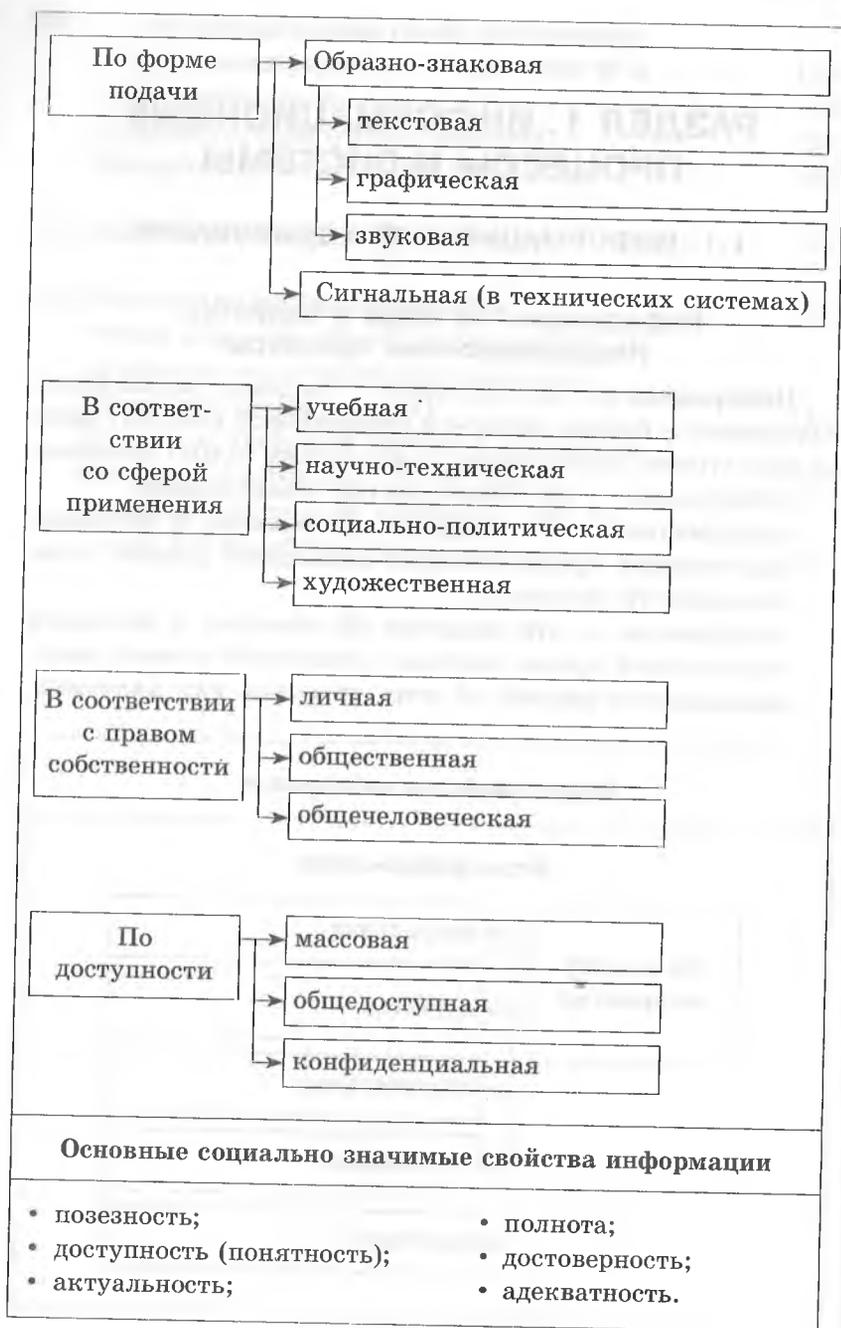
Информация (от лат. *informatio* — сведения, разъяснение, изложение) — базовое понятие в информатике, которому нельзя дать строгое определение — его можно только пояснить:

- информация — это новые факты, новые знания;
- информация — это сведения об объектах и явлениях окружающей среды, которые повышают уровень осведомленности человека;
- информация — это сведения об объектах и явлениях окружающей среды, которые уменьшают степень неопределенности знаний об этих объектах или явлениях.

Виды и свойства информации

Виды информации





Информационные процессы

В современном обществе непрерывно протекают **информационные процессы**: люди воспринимают информацию об окружающем мире с помощью органов чувств, осмысливают ее и принимают определенные решения, которые, воплощаясь в реальные действия, воздействуют на окружающий мир. **Информационный процесс** — это процесс сбора (приема), обработки (преобразования), передачи (обмена) и хранения информации.

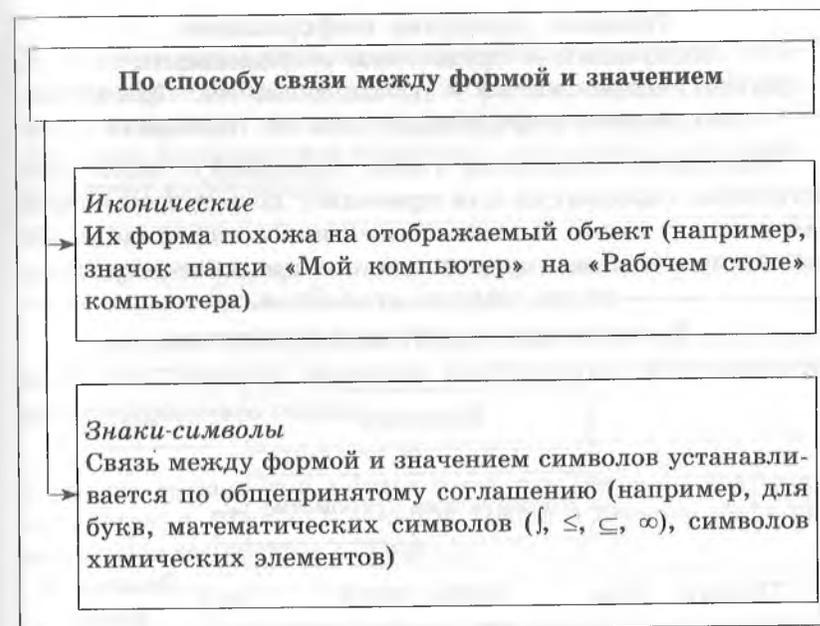
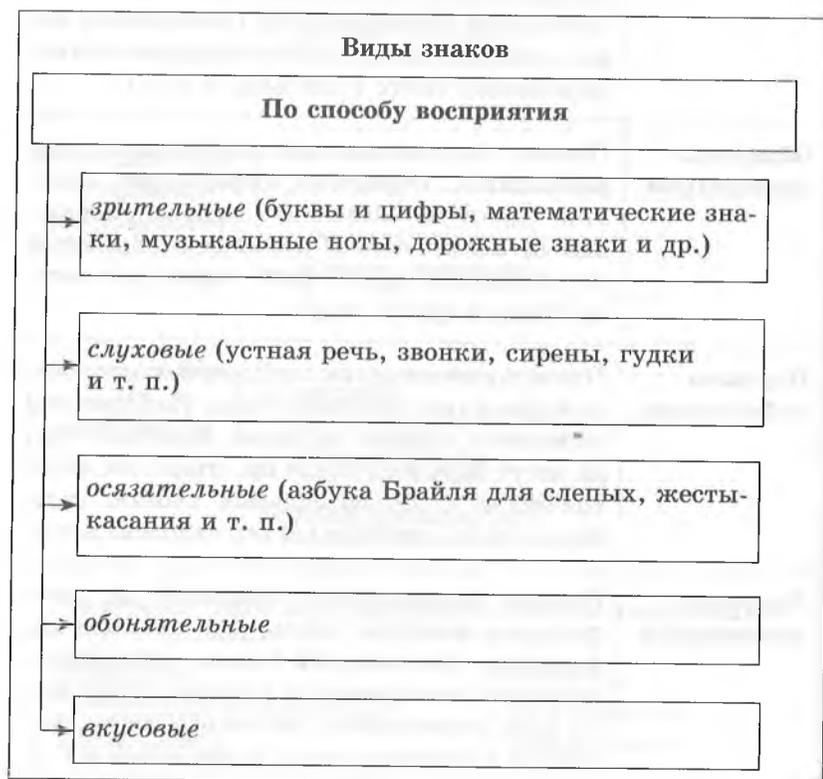
Основные информационные процессы

Сбор информации	Процесс поиска и отбора необходимых сообщений из различных источников (работа со специальной литературой, справочниками; проведение экспериментов; наблюдения; опрос, анкетирование; поиск в информационно-справочных сетях и системах и т. д.)
Обработка информации	Процесс получения новых сообщений из уже имеющихся. Обработка информации является одним из основных способов увеличения ее количества. В результате обработки из сообщения одного вида можно получить сообщения других видов
Передача информации	Процесс перемещения сообщений от источника к приемнику по каналу связи. Информация передается в форме сигналов. Каналами связи могут быть воздушное пространство, электрические и оптоволоконные кабели, отдельные люди, нервные клетки человека и т. д.
Хранение информации	Процесс фиксирования сообщений на материальном носителе. Сейчас для хранения информации используются бумага, деревянные, тканевые, металлические и другие поверхности, кино- и фотопленки, магнитные ленты, магнитные и лазерные диски, флэш-карты и т. д.

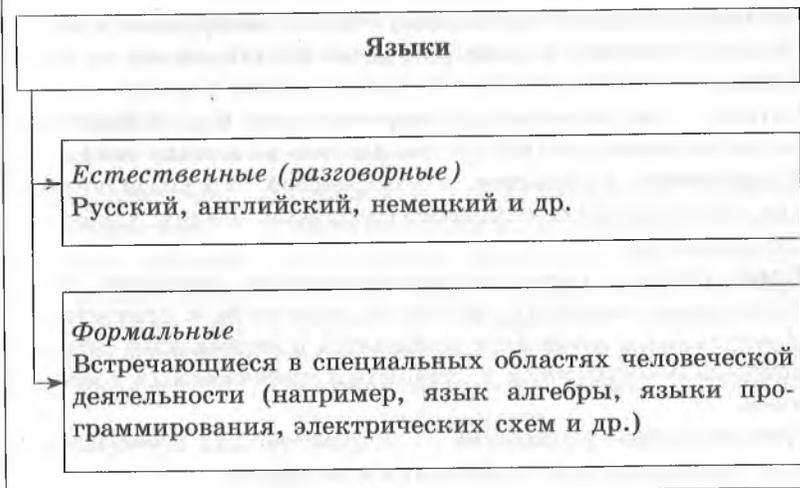
Защита информации	Процесс создания условий, которые не допускают случайной потери, повреждения информации, несанкционированного доступа к ней или ее изменения при хранении. Способами защиты информации являются создание ее резервных копий, хранение в защищенном помещении, предоставление пользователям соответствующих прав доступа к информации, шифрование сообщений и т. д.
--------------------------	--

Язык как способ представления и передачи информации

Для представления, хранения и передачи информации используются знаки.



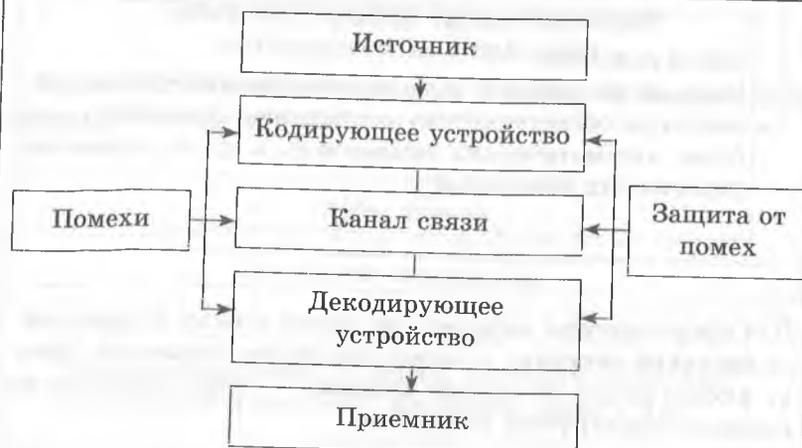
Для представления информации, кроме знаков, используются **знаковые системы**, которые называются языками. Основу любого языка составляет **алфавит** — набор символов, из которых формируется сообщение.



**Процесс передачи информации.
Источники и приемники информации.
Сигнал, кодирование и декодирование. Причины
искажения информации при ее передаче**

Информация передается в виде сообщений от некоторого источника информации к ее приемнику посредством канала связи между ними. В качестве источника информации может выступать живое существо или техническое устройство.

Схема передачи и получения информации



Источник посылает сообщение, которое *кодируется* в передаваемый сигнал, в конечном итоге поступающий на приемник.

Сигнал — это материально-энергетическая форма представления информации (сигнал посылается по каналу связи).

Кодирующее устройство — устройство, предназначенное для преобразования исходного сообщения к виду, удобному для передачи.

Канал связи — совокупность технических устройств, обеспечивающих передачу сигнала от источника к приемнику. В результате в приемнике появляется принимаемый сигнал, который *декодируется* и становится принимаемым сообщением.

Декодирующее устройство — устройство для преобразования закодированного сообщения в исходное.

В процессе передачи информация может теряться и искажаться. Ошибки при передаче данных по телеграфу, атмосферные явления, мешающие прохождению видео- и радиосигналов, неисправность техники — эти *помехи*, или *шумы*, искажают информацию.

Скорость передачи информации и пропускная способность канала связи

Количество информации, передаваемое за единицу времени, есть **скорость передачи информации**, или **скорость информационного потока**.

Скорость выражается в битах в секунду (бит/с) и кратных им Кбит/с и Мбит/с, а также в байтах в секунду (байт/с) и кратных им Кбайт/с и Мбайт/с.

Пропускная способность канала — максимальная скорость передачи информации по каналу связи.

**Методы измерения количества информации:
алфавитный и вероятностный**

Единицей измерения количества информации является бит. **1 бит** — это количество информации, содержащейся в сообщении, которое вдвое уменьшает неопределенность знаний о чем-либо.

Связь между количеством возможных событий N и количеством информации I определяется **формулой Хартли**:

$$N = 2^I$$

Пример. В одной из четырех коробок находится шарик. Таким образом, имеется четыре равновероятных события ($N = 4$). Тогда, согласно формуле Хартли, $4 = 2^I$. Отсюда $I = 2$. Следовательно, сообщение о том, в какой именно коробке находится шарик, содержит 2 бита информации

Алфавитный подход

При алфавитном подходе к определению количества информации отвлекаются от содержания (смысла) информации и рассматривают ее только как последовательность знаков определенной знаковой системы.

Набор символов языка (алфавит) можно рассматривать как различные возможные события. Тогда, если считать, что появление символов в сообщении равновероятно, по формуле Хартли можно рассчитать, какое количество информации несет каждый символ:

$$I = \log_2 N$$

Если N не является целой степенью числа 2, то $\log_2 N$ не является целым числом и для I следует выполнять округление в большую сторону. При решении задач для такого случая I можно найти как $\log_2 N'$, где N' — ближайшая к N степень числа 2, такая, что $N' > N$.

Если количество символов алфавита равно N , а количество символов в записи сообщения равно M , то информационный объем данного сообщения вычисляется по формуле

$$I = M \cdot \log_2 N$$

Пример 1. В русском языке 32 буквы (буква ё обычно не используется), т. е. количество событий будет равно 32. Тогда информационный объем одного символа составит:

$$I = \log_2 32 = 5 \text{ битов.}$$

Пример 2. В английском языке 26 букв. Информационный объем одного символа можно найти следующим образом:

$$\begin{aligned} N &= 26; \\ N' &= 32; \\ I &= \log_2 N' = \\ &= \log_2(2^5) = 5 \text{ битов} \end{aligned}$$

Вероятностный подход

Вероятностный подход к измерению количества информации применяют, когда возможные события имеют различные вероятности реализации.

При вероятностном подходе количество информации определяют по формуле Шеннона:

$$I = -\sum_{i=1}^N p_i \log_2 p_i,$$

где I — количество информации;
 N — количество возможных событий;
 p_i — вероятность i -го события

Пример. Пусть при бросании несимметричной четырехгранной пирамидки вероятности падения на каждую из граней составляют:

$$p_1 = \frac{1}{2}, p_2 = \frac{1}{4}, p_3 = \frac{1}{8}, p_4 = \frac{1}{8}.$$

Тогда количество информации, которое будет получено после реализации одного из случаев, можно вычислить по формуле Шеннона:

$$\begin{aligned} I &= -\left(\frac{1}{2} \cdot \log_2 \frac{1}{2} + \frac{1}{4} \cdot \log_2 \frac{1}{4} + \right. \\ &\quad \left. + \frac{1}{8} \cdot \log_2 \frac{1}{8} + \frac{1}{8} \cdot \log_2 \frac{1}{8} \right) = \\ &= \frac{14}{8} \text{ битов} = 1,75 \text{ бита} \end{aligned}$$

Единицы измерения количества информации

Бит — наименьшая единица измерения информации.

Бит — это количество информации, необходимое для однозначного определения одного из двух равновероятных событий. Например, 1 бит информации получает человек, когда узнает, опаздывает с прибытием нужный ему поезд или нет, был ночью мороз или нет, присутствует на лекции студент Иванов или нет, и т. д.

В информатике принято рассматривать последовательности длиной 8 бит. Такая последовательность называется байтом.

Производные единицы измерения информации:

$$1 \text{ байт} = 8 \text{ бит}$$

$$1 \text{ килобайт (Кб)} = 1024 = 2^{10} \text{ байт}$$

$$1 \text{ мегабайт (Мб)} = 1024 \text{ килобайт} = 2^{20} \text{ байт}$$

$$1 \text{ гигабайт (Гб)} = 1024 \text{ мегабайт} = 2^{30} \text{ байт}$$

$$1 \text{ терабайт (Тб)} = 1024 \text{ гигабайт} = 2^{40} \text{ байт}$$

Представление информации в компьютере. Кодирование чисел

Формат с фиксированной запятой

Целые числа в памяти компьютера хранятся в формате с **фиксированной запятой**: каждому разряду ячейки памяти соответствует один и тот же разряд числа, «запятая» находится вне разрядной сетки.

Если для хранения **целых неотрицательных чисел** отводится **8 бит** памяти. Минимальное число соответствует восьми нулям, хранящимся в восьми битах ячейки памяти, и равно нулю. Максимальное число соответствует восьми единицам и равно:

$$1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + \\ + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 255_{10}$$

Для n -разрядного представления диапазон чисел будет составлять от 0 до $2^n - 1$.

Для хранения **целых чисел со знаком** отводится **2 байта** памяти (16 битов). Старший разряд отводится под знак числа: если число положительное, то в знаковый разряд записывается 0, если число отрицательное — записывается 1. Такое представление чисел в компьютере называется **прямым кодом**.

Для представления **отрицательных чисел** используется **дополнительный код**. Он позволяет заменить арифметическую операцию вычитания операцией сложения, что существенно упрощает работу процессора и увеличивает его быстродействие. Дополнительный код отрицательного числа A , хранящегося в n ячейках, равен $2^n - |A|$.

Алгоритм получения дополнительного кода отрицательного числа

1. Записать прямой код числа в n двоичных разрядах.
2. Получить обратный код числа. Обратный код образуется из прямого кода путем замены нулей единицами, а единиц — нулями, кроме цифр знакового разряда. Обратный код для положительных чисел совпадает с прямым и используется как промежуточное звено для получения дополнительного кода.
3. Прибавить единицу к полученному обратному коду.

Формат с плавающей запятой

Вещественные числа хранятся и обрабатываются в компьютере в формате с **плавающей запятой**, использующем экспоненциальную форму записи чисел.

Число в экспоненциальной форме представляется в виде

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

где m — мантисса числа (правильная, отличная от нуля дробь);

q — основание системы счисления;

n — порядок числа.

Число в формате с плавающей запятой может занимать в памяти 4 байта (*обычная точность*) или 8 байт (*двойная точность*). При записи числа выделяются разряды для хранения знака мантиссы, знака порядка, а также порядка и мантиссы. Две последние величины определяют диапазон изменения чисел и их точность.

Кодирование текстовой информации

Кодирование символа — это присвоение символу конкретного числового кода.

При вводе в компьютер текстовой информации происходит ее двоичное кодирование.

Код символа хранится в оперативной памяти компьютера. В процессе вывода символа на экран производится обратная операция — декодирование, т. е. преобразование кода символа в его изображение.

Как правило, для хранения кода символа используется 1 байт (8 бит), поэтому коды символов могут принимать значения от 0 до 255. Такие кодировки называют **однобайтными**. Они позволяют использовать 256 символов ($N = 2^8 = 256$).

Таблица однобайтных кодов символов называется ASCII (American Standard Code for Information Interchange — Американский стандартный код для обмена информацией).

Первая часть таблицы ASCII-кодов (от 0 до 127) одинакова для всех IBM-PC-совместимых компьютеров и содержит:

- коды управляющих символов;
- коды цифр, арифметических операций, знаков препинания;
- некоторые специальные символы;
- коды больших и маленьких латинских букв.

Вторая часть таблицы ASCII (коды от 128 до 255) бывает различной в разных компьютерах. Она содержит коды букв национального алфавита, коды некоторых математических символов, коды символов псевдографики. Для русских букв в настоящее время имеется пять различных кодовых таблиц: КОИ-8, CP1251, CP866, Mac, ISO.

Широкое распространение в последнее время получил новый международный стандарт Unicode. В нем отводится по два байта (16 бит) для кодирования каждого символа, поэтому с его помощью можно закодировать 65536 различных символов ($N = 2^{16} = 65536$). Коды символов могут принимать значения от 0 до 65535.

Пример. С помощью кодировки Unicode закодирована фраза: *Я хочу поступить в университет.*

Нужно определить информационный объем этой фразы.

Решение. В данной фразе содержится 31 символ (включая пробелы и знак препинания). Поскольку в кодировке Unicode каждому символу отводится 2 байта памяти, для всей фразы понадобится $31 \cdot 2 = 62$ байта, или $31 \cdot 2 \cdot 8 = 496$ бит.

Ответ: 32 байта, или 496 бит.

1.2. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Общее понятие об алгоритме

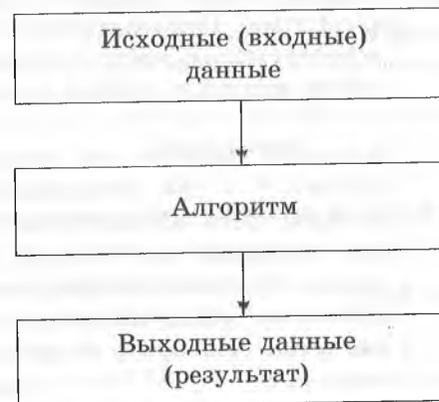
Алгоритм — это точное и полное описание последовательности действий над заданными объектами, позволяющее получить конечный результат.

Процесс построения алгоритма

Процесс построения алгоритма (алгоритмизация) — разложение задачи на элементарные действия или операции.

Определение алгоритма для применения в области информатики нуждается в уточнении путем введения некоторых дополнительных ограничений.

1. Решение задач в информатике всегда связано с преобразованием информации, а значит, исходными данными и результатом работы алгоритма должна быть информация.



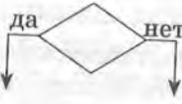
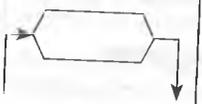
2. Алгоритмы в информатике предназначены для реализации в виде компьютерных программ или для создания некоторой компьютерной технологии. Для выполнения алгоритма требуется конечный объем оперативной памяти и конечное время.

Основные требования, предъявляемые к алгоритмам	
Дискретность (прерывность)	Алгоритм должен представлять решение задачи в виде последовательности простых (или ранее определенных) этапов (шагов). Каждый шаг алгоритма формулируется в виде инструкций (команд)
Определенность	Шаги (операции) алгоритма должны допускать однозначную трактовку и быть понятными для исполнителя алгоритма. Это свойство указывает на то, что любое действие в алгоритме должно быть строго определено и описано для каждого случая
Массовость	Алгоритм должен давать решение не только для конкретного набора данных, но и для целого класса задач, который определяется диапазоном возможных исходных данных (область применимости алгоритма). Требование массовости подразумевает использование переменных в качестве исходных данных алгоритма
Результативность	Алгоритм нацелен на конкретный результат, т. е. при выполнении алгоритма должны быть рассмотрены все возможные ситуации, для каждой из которых должен быть получен результат. Под результатом может пониматься и сообщение о том, что задача не имеет решения.
Конечность	Количество шагов алгоритма должно быть конечным
Эффективность	Количество шагов и сами шаги алгоритма должны быть такими, чтобы решение могло быть найдено за конечное и, кроме того, приемлемое время

Формы представления алгоритмов

Наиболее распространенные методы (языки) для описания алгоритмов	
Обычный язык	Алгоритм излагается на обычном языке с разделением на последовательные шаги
Блок-схема	Графическое изображение алгоритма с помощью специальных значков-блоков
Псевдокод	Синтез алгоритмического и обычного языков. Элементы некоторого базового алгоритмического языка используются для строгой записи базовых структур алгоритма
Формальный алгоритмический язык (язык программирования)	При записи алгоритмов используют строго определенный набор символов и составленных из них специальных зарезервированных слов. Имеются строгие правила построения языковых конструкций

Блок-схема		
Наиболее часто употребляемые блочные символы		
Название символа	Графическое изображение	Комментарии
Пуск/Останов (блоки начала и конца алгоритма)		Указание на начало или конец алгоритма
Ввод/Вывод данных (блоки ввода, вывода)		Организация ввода/вывода в общем виде

Процесс (операторные блоки)		Выполнение вычислительного действия или последовательности действий (можно объединять в один блок), которые изменяют значение, форму представления или размещение данных
Условие (условный блок)		Выбор направления выполнения алгоритма. Если условие, записанное внутри ромба, выполняется, то управление передается по стрелке «да», в противном случае — по стрелке «нет». Таким образом, реализуется процесс изменения последовательности вычислений в зависимости от выполнения условия
Начало цикла с параметром		Используется для организации циклических конструкций с известным количеством <i>итераций</i> (повторений) и известным шагом изменения параметра цикла. Внутри блока для параметра цикла указываются через запятую его начальное и конечное значение, конечное значение и шаг изменения. Цикл, для которого неизвестно количество повторений, записывается с помощью условного и операторных блоков

Предопределенный процесс		Используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращения к библиотечным подпрограммам
Печать сообщений		Вывод результатов на печать

Условия, которые требуют проверки при составлении блок-схемы

Из каждого прямоугольника и параллелограмма (кроме конца алгоритма) должна выходить только одна стрелка	В каждый прямоугольник и параллелограмм (кроме начала алгоритма) должна входить хотя бы одна стрелка	В каждый ромб должна входить хотя бы одна стрелка, а выходить из него — две стрелки, помеченные словами «да» и «нет»
---	--	--

Псевдокод

Псевдокод занимает промежуточное положение между естественным языком и языками программирования. В отличие от формальных языков, в псевдокоде не приняты строгие синтаксические правила для записи команд. Однако в псевдокоде есть конструкции, которые присущи формальным языкам.

Одним из примеров псевдокода является школьный алгоритмический язык АЯ. Алфавит этого учебного алгоритмического языка открытый. В него могут быть введены любые понятные всем символы: русские и латинские буквы, знаки математических операций, знаки отношений, специальные знаки и т. п. Кроме алфавита, в алгоритмической нотации определяются *служебные слова*, которые являются неделимыми.

Некоторые служебные слова учебного алгоритмического языка

алг — заголовок алгоритма	нц — начало цикла	знач
нач — начало алгоритма	кц — конец цикла	и
кон — конец алгоритма	дано	или
арг — аргумент	надо	не
рез — результат	если	да
цел — целый	то	нет
сим — символьный	иначе	при
лит — литерный	всё	выбор
лог — логический	пока	ввод
вещ — вещественный	для	вывод
таб — таблица	от	
длин — длина	до	

Общий вид записи алгоритма на псевдокоде

алг — название алгоритма (аргументы и результаты)
 дано — условие применимости алгоритма
 надо — цель выполнения алгоритма
 нач — описание промежуточных величин, последовательность команд (тело алгоритма)
 кон

Часть алгоритма от слова алг до слова нач называется *заголовком*, а часть, заключенная между словами нач и кон, — *телом алгоритма* (исполняемой частью алгоритма).

В предложении алг после названия алгоритма в круглых скобках указываются *характеристики* (арг, рез) и *тип*

значения (цел, вещ, сим, лит или лог) всех входных (аргументы) и выходных (результаты) переменных. При описании массивов (таблиц) используется служебное слово таб, дополненное именем массива и граничными парами по каждому индексу элементов массива.

Пример записи алгоритма на псевдокоде

алг Сумма квадратов целых чисел до n включительно
 (арг цел n рез цел S)

дано $n > 0$

надо $S = 1 * 1 + 2 * 2 + 3 * 3 + \dots + n * n$

нач

цел i

ввод n ;

$S := 0$

нц для i от 1 до n

$S := S + i * i$

кц

вывод "S = ", S

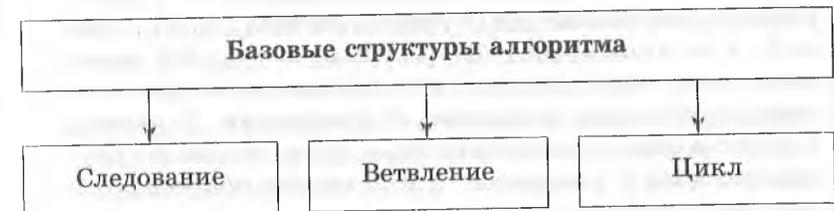
кон

Примечание. В предложениях дано и надо после знака «|» записаны комментарии.

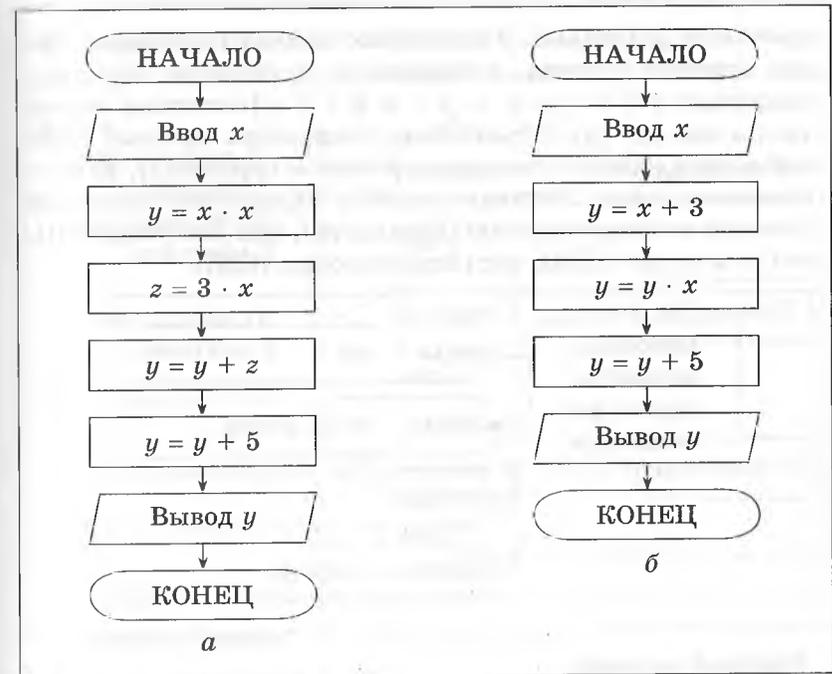
Комментарии можно помещать в конце любой строки — они существенно облегчают понимание алгоритма.

Использование основных алгоритмических конструкций: следование, ветвление, цикл

Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур.



Базовая структура СЛЕДОВАНИЕ	
Структура Следование указывает на то, что управление передается последовательно от одного действия к другому.	
Учебный алгоритмический язык	Язык блок-схем
действие 1 действие 2 ... действие n	
Использование исключительно этой структуры возможно только для достаточно простых задач, ход решения которых не меняется при изменении конкретных исходных данных и состоит в последовательном выполнении определенных операций.	
<p>Пример. Найти $y(x) = x^2 + 3x + 5$, пользуясь только операциями умножения и сложения.</p> <p>Решение. На рисунках <i>a</i> и <i>б</i> приводятся два алгоритма, реализующие решение поставленной задачи.</p> <p>Порядок вычисления $y(x)$ в первом случае (рис. <i>a</i>) — обычный, а во втором (рис. <i>б</i>) — $(x + 3)x + 5$. Оба порядка вычисления эквивалентны, но в первом случае для вычисления необходимо выполнить 2 умножения, 2 сложения и использовать 3 переменных (x, y, z); во втором же случае выполняются 1 умножение и 2 сложения при использовании двух переменных (x, y).</p>	



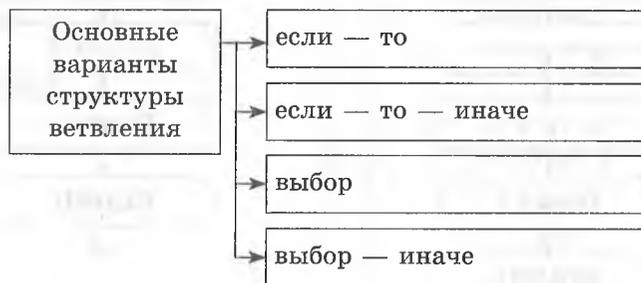
Базовая структура ВЕТВЛЕНИЕ (РАЗВИЛКА)

Структура Ветвление используется в том случае, когда в зависимости от результата проверки условия выполнение программы может измениться и пойти двумя разными (альтернативными) путями.

Различают две структуры этого типа — *полную* и *неполную*. Для полной структуры, если условие выполняется (является истинным), вслед за ним выполняется действие 1, иначе — действие 2. В случае неполной структуры, если условие выполняется (является истинным), вслед за ним выполняется действие 1, иначе — ничего не происходит.

Важную роль в операторах ветвления играют содержащиеся в них условия. В простейшем случае условиями служат отношения между величинами. Условия с одним отношением называются *простыми условными выражениями*, или

простыми условиями. В некоторых задачах необходимы более сложные условия, состоящие из нескольких простых, например: $a < x < c$, т. е. $x < a$ и $x > c$ (возможна запись « $x < a$ and $x > c$ »). Объединение нескольких простых условий в одно образует *составное условное выражение*, или *составное условие*. Составные условия образуются с помощью логических операторов **not** (отрицание), **and** (логическое И), **or** (логическое ИЛИ), **xor** (исключающее ИЛИ).



Учебный алгоритмический язык

Язык блок-схем

1) если — то

если условие
то действие 1
всё



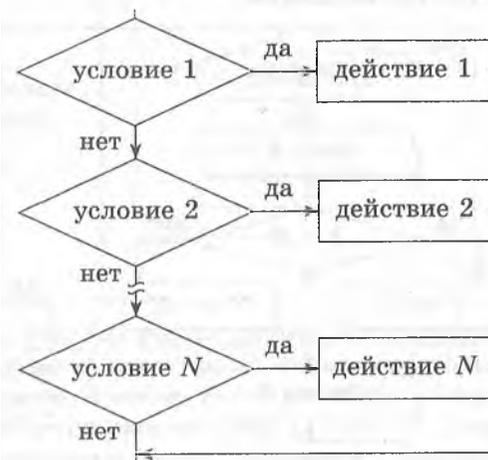
2) если — то — иначе

если условие
то действие 1
иначе действие 2
всё



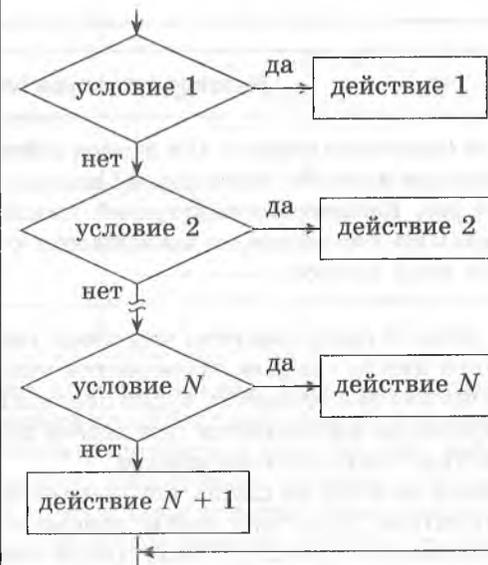
3) выбор

выбор
при условии 1:
действие 1
при условии 2:
действие 2
...
при условии N:
действие N
всё

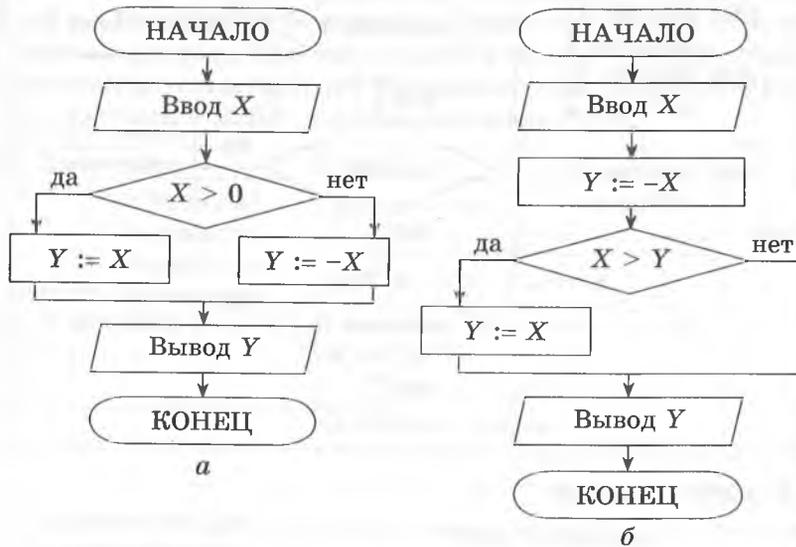


4) выбор — иначе

выбор
при условии 1:
действие 1
при условии 2:
действие 2
...
при условии N:
действие N
иначе действие
N + 1
всё



Пример. Рассмотрим нахождение модуля числа $y(x) = |x|$. Решение приведено для случаев полной (а) и неполной (б) структур ветвления.



Базовая структура ЦИКЛ

Эта структура служит для записи алгоритмов, в которых некоторая их часть (тело цикла) должна повторяться несколько раз. Количество повторений цикла может определяться разными способами, в зависимости от которых различают три вида циклов.

1. Цикл с предусловием, или цикл «пока». При реализации этого цикла сначала проверяется условие его выполнения. Пока оно выполняется, будет повторяться тело цикла. Если условие не выполняется при первой проверке, то тело цикла не будет выполняться вообще. После выхода из цикла управление передается следующей структуре. Для того чтобы избежать заикливания, т. е. бесконечного цикла, в теле цикла обязательно должны изменяться записанные в условии параметры.

Учебный алгоритмический язык	Язык блок-схем
<p>нц пока условие тело цикла (последовательность действий) кц</p>	
<p>2. Цикл с параметром. Этот вид цикла удобно использовать в тех случаях, когда заранее известно количество повторений цикла. Здесь вводится понятие <i>счетчик цикла</i>, который по умолчанию считается равным либо 1, либо -1. Для организации цикла необходимо задать верхнюю и нижнюю границы изменения счетчика цикла.</p>	
Учебный алгоритмический язык	Язык блок-схем
<p>нц для i от i_1 до i_2 тело цикла (последовательность действий) кц</p>	
<p>3. Цикл с постусловием, или цикл «до». При реализации этого цикла условие проверяется после тела цикла, и таким образом, тело цикла всегда выполняется хотя бы один раз. Цикл будет выполняться <i>до</i> выполнения условия, отсюда и его название — <i>цикл «до»</i>. А пока условие не выполнено, будет повторяться тело цикла (выполнение условия, таким образом, является условием окончания цикла).</p>	

В этом случае, как и в цикле «пока», необходимо предусмотреть в теле цикла изменение параметров условия цикла.

Учебный алгоритмический язык

Язык блок-схем

нц
тело цикла (последовательность действий)
до условие
кц



Итерационный цикл. Особенностью итерационного цикла является то, что число повторений операторов тела цикла заранее неизвестно. Для его организации используется цикл типа *пока*. Выход из итерационного цикла осуществляется в случае выполнения заданного условия. На каждом шаге вычислений происходит последовательное приближение и проверка условия достижения искомого результата.

Пример. Вычислить сумму знакопеременного ряда $S = x -$

$$-\frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$
 с заданной точностью.

Для данной знакопеременной бесконечной суммы требуемая точность будет достигнута, когда очередное слагаемое по абсолютной величине станет меньше ε .

Решение. Запишем блок-схему алгоритма, в которой будем использовать следующие обозначения:

S — частичная сумма ряда (стартовое значение равно 0);

ε — точность вычисления;

i — номер очередного слагаемого;

m — значение очередного слагаемого;

p — числитель очередного слагаемого.

Учебный алгоритмический язык

Язык блок-схем

алг Сумма (арг вещь X , ε , рез вещь S)

дано $0 < X < 1$

надо $S = X - X^2/2 + X^3/3 - X^4/4 + \dots$

нач цел I , вещь M, P

ввод X, ε

$S := 0; I := 1$

$M := 1; P := -1$

нц пока $\text{abs}(M) > \varepsilon$

$P := -P * X$

$M := P/I$

$S := S + M$

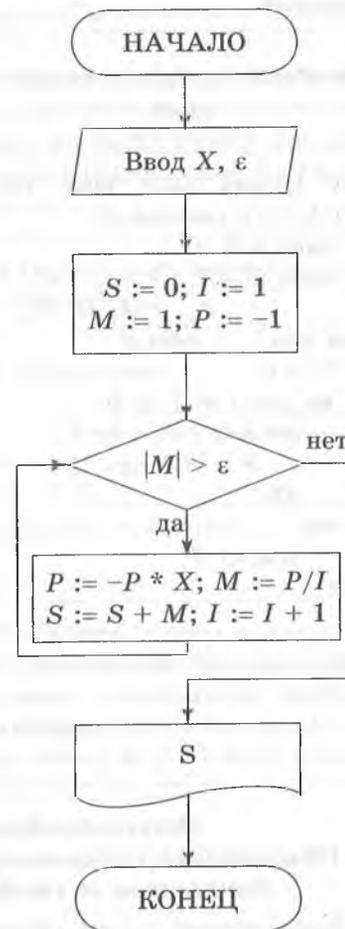
$I := I + 1$

кц

вывод S

кон

Примечание. Следует отметить, что ряд будет сходящимся только при выполнении условия $0 < x < 1$



Вложенные циклы. Возможны случаи, когда внутри тела цикла необходимо повторить некоторую последовательность операторов, т. е. организовать внутренний цикл. Такая структура получила название *цикл в цикле*, или *вложенный цикл*. Глубина вложения циклов (количество вложенных друг в друга циклов) может быть различной.

Пример. Вычислить сумму элементов для заданной матрицы $A(5,3)$ (таблицы из пяти строк и трех столбцов).

Решение

Учебный алгоритмический язык	Основная часть блок-схемы нахождения суммы элементов матрицы
<p>алг Сумма (арг вещ таб $A(1:5,1:3)$, рез вещ S) дано таб A надо $S = A_{1,1} + A_{1,2} + A_{1,3} + \dots + A_{5,1} + A_{5,2} + A_{5,3}$ нач цел i, j, вещ S $S := 0$ нц для i от 1 до 5 нц для j от 1 до 3 $S := S + A[i, j]$ кц кц вывод S кон</p>	

Здесь порядок выполнения вложенных циклов следующий: счетчик внутреннего цикла изменяется быстрее, т. е. для $i = 1$ (внешний цикл) j пробегает значения 1, 2, 3 (внутренний цикл); далее $i = 2$, и j опять пробегает значения 1, 2, 3; и т. д.

Использование переменных.

Объявление переменной (тип, имя, значение).

Локальные и глобальные переменные

Переменная — это объект программы, имеющий имя и изменяемое значение. Для хранения переменной в памяти компьютера выделено определенное место.



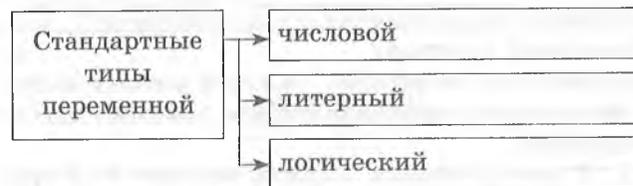
Каждая переменная имеет свой **идентификатор** — свое уникальное имя. Имя переменной указывает, в каком месте памяти компьютера она хранится. Двух переменных с одним именем в одном программном модуле (блоке) быть не должно.

Имена составляются из букв и цифр, первой литерой должна быть буква. Разумно давать переменным логически осмысленные имена, в соответствии с их назначением. Нельзя использовать в качестве имен зарезервированные (служебные) слова языка программирования.

Тип переменной — это диапазон (набор) всех значений, которые может принимать данная переменная.

Тип переменной определяет:

- необходимый размер памяти
- диапазон значений, которые может принимать величина
- возможные операции над величиной
- формы представления величин (или формат представления величин)



Значение переменной — это непосредственно то, чему равна данная переменная в конкретный момент времени. Это может быть число, символ, текст и т. д. Значение переменной в программе можно задать двумя способами: с помощью операции присваивания и с помощью процедуры ввода.

Работа с массивами

Векторы и таблицы в программировании принято называть массивами.

Массив — это упорядоченный набор данных, имеющий одно имя и состоящий из фиксированного числа однотипных элементов. Каждый элемент массива снабжен индексом.
Индекс — это порядковый номер элемента, определяющий его положение в массиве. Таким образом, элементы массива идентифицируются с помощью имени массива и своих индексов. Количество элементов массива называется **размерностью массива**.

Основные характеристики массива

→ размерность

→ базовый тип элементов

→ тип индекса (может быть только порядковый тип)

→ множество значений для индекса

Виды массивов

Одномерный (вектор)

→ Содержит одно измерение, каждый элемент массива обозначается идентификатором (именем) массива с индексом

Многомерный (матрица)

→ Содержит n -е количество измерений; каждый элемент массива обозначается идентификатором массива для n индексов

Способы заполнения массива данными

→ непосредственное присваивание значений элементам

→ заполнение массива произвольными элементами или случайными числами

→ ввод значений элементов с клавиатуры или чтение из файла

Обработка массивов

Задачи, связанные с обработкой массивов, условно можно разделить на три вида:

1. Задачи, решаемые «в одно действие»
2. Стандартные задачи
3. Задачи, решение которых требует знания вспомогательных алгоритмов, специальных методов и приемов.

Примеры решения простейших задач (для одномерных массивов)

Пример 1. В массиве a каждый элемент равен 0 или 1. Заменить все нули единицами и наоборот.

Решение. Достаточно одного оператора присваивания в теле цикла:

$$a[i] := 1 - a[i].$$

Пример 2. В массиве каждый элемент равен 0, 1 или 2. Переставить элементы массива так, чтобы сначала располагались все нули, затем все единицы и, наконец, все двойки. Дополнительный массив не использовать.

Решение. Можно не переставлять элементы массива, а подсчитать количество нулей, единиц, двоек и заново заполнить массив требуемым образом.

Пример 3. Даны два n -элементных массива x и y одного типа. Поменять местами все x_i и y_i , ($i = 1 \dots n$), не используя промежуточные величины.

Решение. Перестановку можно выполнить в цикле для всех i от 1 до n с помощью серии из трех операторов присваивания:

$$x[i] := x[i] + y[i];$$

$$y[i] := x[i] - y[i];$$

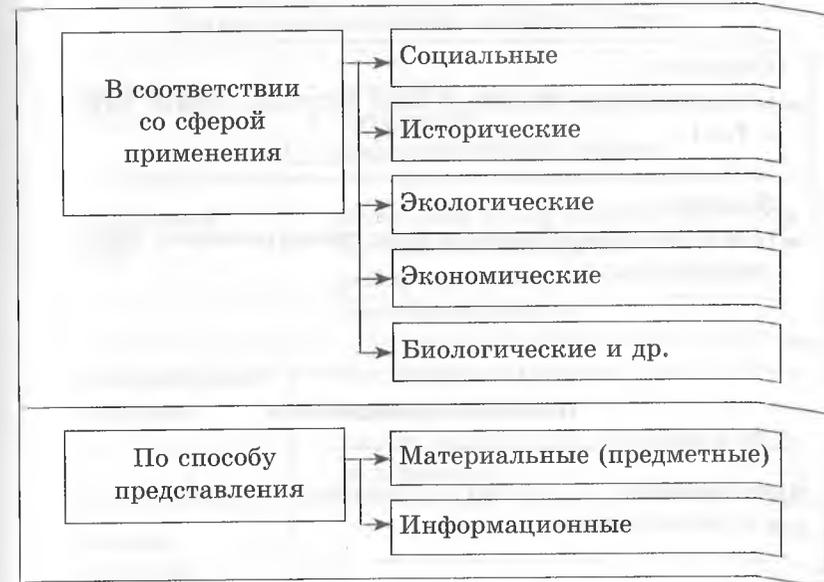
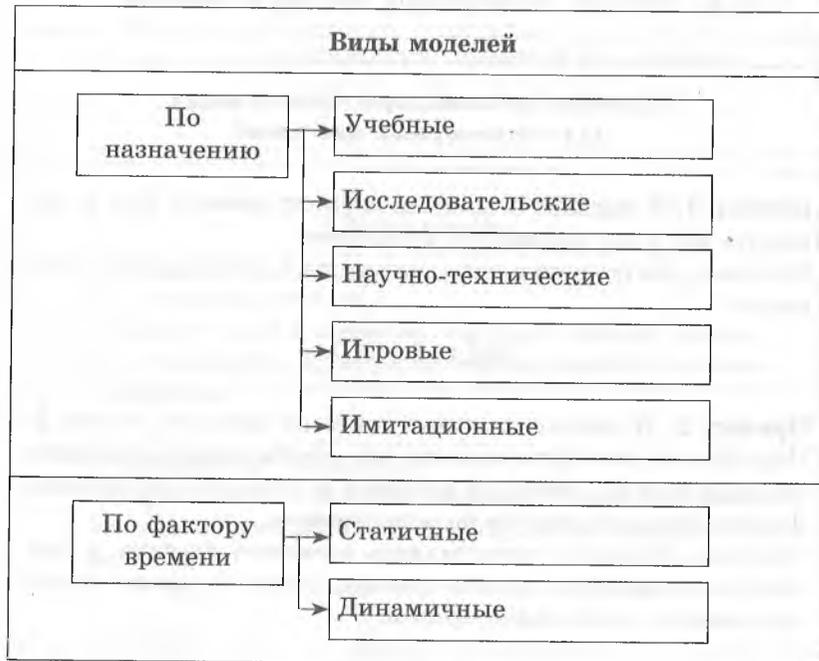
$$x[i] := x[i] - y[i].$$

1.3. МОДЕЛИРОВАНИЕ И КОМПЬЮТЕРНЫЙ ЭКСПЕРИМЕНТ

Модели. Виды моделей. Понятие моделирования

Модель — это объект, сохраняющий существенные черты изучаемого объекта и заменяющий его во время исследований.

Исследуемый объект по отношению к модели является оригиналом, образцом, прототипом.



Информационные модели

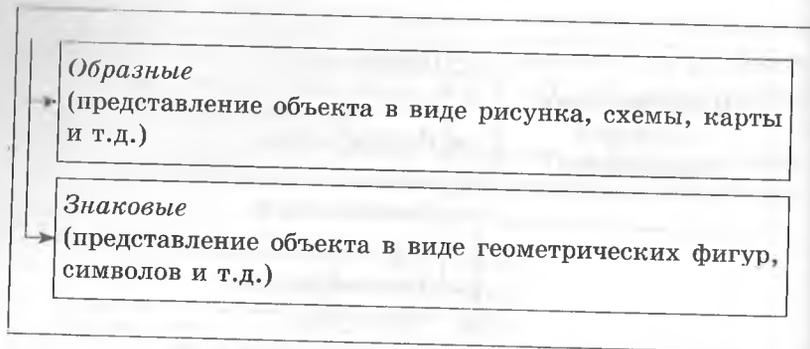
Информационная модель — это совокупность целенаправленно выбранной информации о свойствах объекта, его состоянии и связи с внешним миром, которая является существенной для данного исследования.

Один и тот же объект при различных исследованиях может быть представлен разными моделями.

Одна и та же модель может служить для исследования разных прототипов.

Виды информационных моделей

- **Воображаемые**
(мысленное представление об объекте, сформированное в сознании человека)
- **Вербальные**
(устное или письменное описание объекта)



Понятие моделирования

Моделирование — это метод познания, состоящий в создании и исследовании моделей.

Основные цели информационного моделирования

- Создание объектов с заданными свойствами
- Определение последствий воздействия на объект и принятие правильного решения
- Прогнозирование поведения объекта моделирования
- Достижение эффективного управления объектом или процессом

Компьютерное моделирование

Компьютерная модель — это модель, реализуемая с помощью программных средств.

Компьютерное моделирование — это исследование свойств объекта путем проведения эксперимента с его компьютерной моделью.

Этапы компьютерного моделирования

1. Постановка задачи	<ul style="list-style-type: none"> • описание • мотивация • предварительный анализ
2. Разработка модели	<ul style="list-style-type: none"> • выделение существенных факторов • составление алгоритма • выбор программного обеспечения • программирование
3. Компьютерный эксперимент	<ul style="list-style-type: none"> • тестирование модели • отладка модели • расчет модели при различных входных данных
4. Анализ результатов	

Применение компьютерного моделирования

- Проведение научных исследований
- Выяснение причин техногенных аварий и природных катастроф
- Подготовка проектно-конструкторской документации
- Создание систем искусственного интеллекта, интеллектуальных обучающих систем
- Создание спецэффектов для кинофильмов и рекламы, разработка компьютерных игр
- Моделирование климатических явлений
- Моделирование ситуации для принятия важных решений (в социально-экономической и др. сферах)

1.4. КОМПЬЮТЕР КАК СРЕДСТВО АВТОМАТИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

Типы компьютеров, их основные характеристики и области использования

Основные типы и характеристики современных компьютеров		
Тип компьютера	Назначение	Быстродействие (операций в секунду)
Мэйнфрейм	Решение задач, требующих больших объемов вычислений (сложные вычисления в аэродинамике, метеорологии, физике высоких энергий; проведение фундаментальных экспериментов и т. д.)	Сотни триллионов
Средний	Серверы предприятий, банков, учреждений, организаций	Сотни миллиардов
Персональный	Обеспечение потребностей отдельного пользователя	До сотен миллионов
Портативный: • ноутбук • карманный	Обеспечение мобильности пользователя, т. е. возможность работать с компьютером в разных местах при отсутствии постоянного источника питания	Десятки миллионов
Встроенные	Автоматизация определенных функций современных устройств бытового и промышленного назначения	От сотен тысяч до десятков миллионов

Архитектура современного компьютера

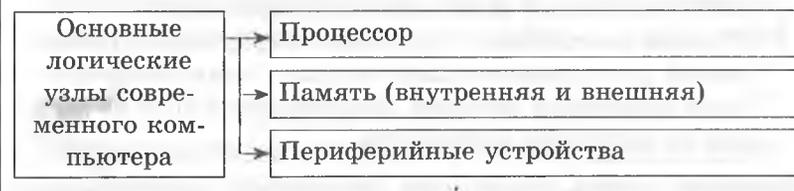
Современный компьютер — сложное электронное устройство, состоящее из нескольких важных функциональных блоков, взаимодействующих между собой.

Под термином архитектура компьютера подразумевается его логическая организация, структура и ресурсы, т. е. средства вычислительной системы, которые могут быть выделены на определенный интервал времени процессу обработки данных.

В основу построения большинства компьютеров положены принципы, сформулированные Джоном фон Нейманом

1. *Принцип программного управления* — программа состоит из набора следующих друг за другом в определенной последовательности команд, которые автоматически выполняются процессором.
2. *Принцип однородности памяти* — программы и данные хранятся в одной и той же памяти.
3. *Принцип адресности* — основная память структурно состоит из пронумерованных ячеек.

Компьютеры, построенные на этих принципах, имеют классическую архитектуру



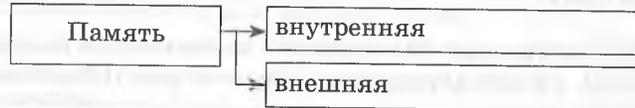
Процессор

Процессор является главным устройством компьютера, в котором происходит обработка всех видов информации. Другой важной функцией процессора является обеспечение согласованного действия всех узлов, входящих в состав компьютера. Наиболее важными частями процессора являются арифметико-логическое устройство (АЛУ) и устройство управления (УУ).

Внутри процессора имеются специальные ячейки (**регистры**) для оперативного хранения обрабатываемых данных и некоторой служебной информации

Память

Память в целом предназначена для хранения данных, а также программ для их обработки.



Под **внутренней** памятью современного компьютера принято понимать быстродействующую электронную память, размещенную на его системной плате. Сейчас такая память изготавливается на базе самых современных полупроводниковых технологий.

Части внутренней памяти

ОЗУ (оперативное запоминающее устройство)

→ Главное назначение — хранение данных и программ для решаемых в текущий момент задач

ПЗУ (постоянное запоминающее устройство)

→ Главное назначение — хранение информации, необходимой для первоначальной загрузки компьютера в момент включения питания. Информация в ПЗУ не зависит от состояния компьютера

Внешняя память реализуется посредством разнообразных устройств для хранения информации, конструктивно оформленных в виде самостоятельных блоков.

Устройства для хранения информации

- Накопители на гибких магнитных дисках
- Накопители на жестких магнитных дисках (винчестеры)
- Оптические накопители (CD-ROM, DVD-ROM)

Периферийные устройства

Периферийные устройства обеспечивают взаимодействие компьютера с объектами окружающей среды: пользователями, объектами управления и другими компьютерами.

Основные периферийные устройства

Устройства ввода информации

- Клавиатура
- Манипуляторы
- Сенсорная панель
- Графический планшет
- Сканер
- Цифровые камеры
- Звуковая карта и микрофон
- Джойстик

Устройства вывода информации

- Монитор
- Принтер
- Плоттер
- Акустические колонки и наушники

Подсоединение периферийных устройств к компьютеру производится через специальные интерфейсы — порты ввода-вывода.

Порты ввода-вывода (по способу передачи информации)

Последовательные

→ Информация передается последовательно, бит за битом; для передачи информации используется один провод (подключаются мышь, модем)

Параллельные

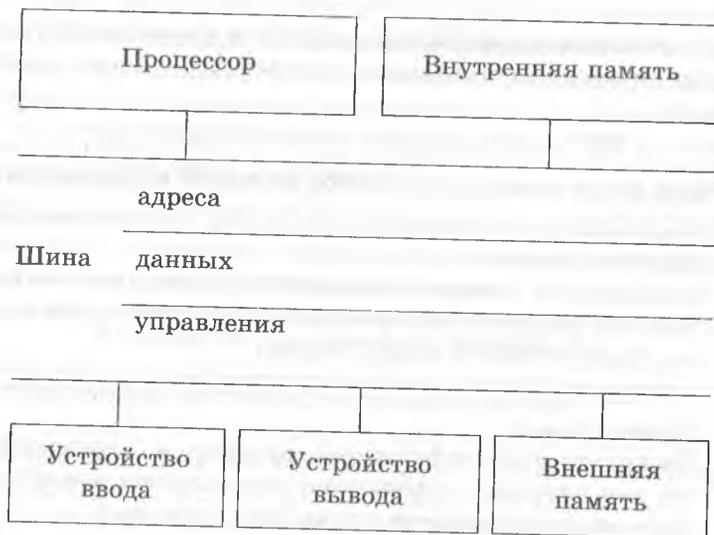
→ Несколько бит информации передается одновременно; для передачи информации используется несколько проводов (подключаются принтеры, сканеры)

В настоящее время эти порты вытесняются шиной USB (**Universal Serial Bus** — универсальная последовательная шина), которая представляет собой последовательный интерфейс передачи данных для средне- и низкоскоростных периферийных устройств. Шина рассчитана на подключение и автоопределение до 127 устройств, а также на подключение к работающему компьютеру без его перезагрузки.

Магистрально-модульный принцип построения компьютера

Архитектура современных персональных компьютеров основана на магистрально-модульном принципе. Модульный принцип позволяет пользователю самому комплектовать нужную ему конфигурацию компьютера и при необходимости производить ее модернизацию.

В основу модульной организации системы положен магистральный принцип обмена информацией. Все контроллеры устройств взаимодействуют с микропроцессором и оперативной памятью через системную магистраль передачи данных, называемую системной шиной.



Файлы и файловые системы

Во всех операционных системах имеющаяся в компьютере информация хранится в виде файлов.

Файл

Файл — это снабженная именем упорядоченная совокупность данных, которую операционная система обрабатывает как единое целое.

Размер файла — объем памяти, который файл занимает на носителе.

Имя файла состоит из двух частей — собственно имени файла и расширения, которые записываются через точку. Например: *реферат.doc*.

Правила присвоения имен файлам

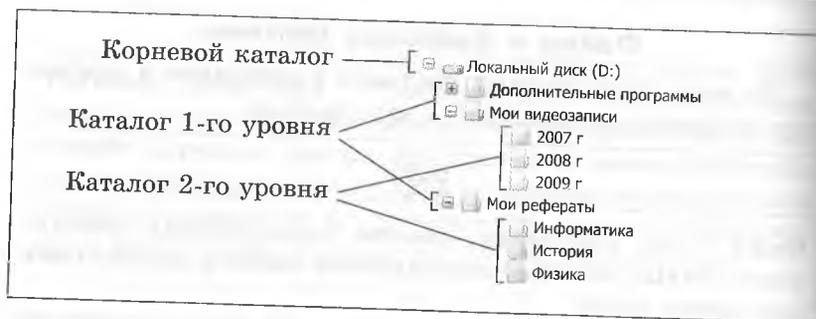
1. Можно использовать латинские буквы, буквы других алфавитов, цифры, пробелы, специальные символы (кроме символов * ? : < > | \ /).
2. Название файла должно характеризовать его смысл;
3. Расширение в имени файла указывает на тип файла, формат записи или программу, в которой он был создан. Каждому типу файлов в системе Windows соответствует своя пиктограмма.

Каталоги

Каталог — это специальное место на диске, где сохраняются имена файлов, сведения об их размере, дате и времени создания и последнего обновления, а также основные свойства файла.

В терминологии ОС Windows каталог называется **папкой**. Требования к имени каталогов (папок) такие же, как и к имени файла, но, как правило, они не имеют расширений.

Корневой каталог — это главный каталог каждого диска. В нем регистрируются файлы и каталоги 1-го уровня. В каталогах 1-го уровня, в свою очередь, регистрируются файлы и каталоги 2-го уровня и т. д. Имена каталогов, включенных один в другой, отделяются символом «/».



Файловая система

Файловая система — это совокупность программ, которые обеспечивают работу с файлами и каталогами, а также сами файлы и каталоги, которые хранятся на устройствах внешней памяти.

Виды файловых систем

Одноуровневая

Применяется для дисков с небольшим количеством файлов и представляет собой линейную последовательность имен файлов

Многоуровневая (иерархическая)

Применяется для дисков с большим количеством файлов и представляет собой совокупность папок, вложенных друг в друга

Путь к файлу

Путь к файлу — это последовательность имен каталогов от корневого до того каталога, в котором находится файл.
Например: C:/Мои документы/Информатика/Реферат.doc
Полное имя файла = путь к файлу + имя файла
Например: C:/Мои документы/Фото/Мой класс/Друзья.bmp

Операции с файлами



Архивация файлов

Для долговременного хранения или передачи по компьютерным сетям целесообразно **архивировать** файлы, т. е. уменьшать их информационный объем.

Архиваторы — специальные программы, используемые для архивации файлов и их последующего восстановления.

Программы-архиваторы

WinZIP, WinRAR, 7-ZIP, Winace, PowerArchiver, ArjFolder, BitZipper, bzip2 и др.

Форматы архивных файлов

ZIP, RAR, ARJ, CAB, LZH, ACE, ISO и др.

Программное обеспечение компьютера

Программное обеспечение — это совокупность программ и сопровождающей их документации, предназначенная для решения определенных задач на компьютере.

Операционная система

Операционная система — необходимая составляющая программного обеспечения компьютера, которая осуществляет организацию вычислительного процесса и управление ресурсами компьютера.

Функции операционной системы

- Контроль работоспособности оборудования компьютерной системы
- Выполнение процедуры начальной загрузки
- Управление работой устройств компьютера
- Взаимодействие пользователя с компьютером
- Загрузка и выполнение прикладных программ
- Распределение ресурсов компьютера (оперативной памяти, процессорного времени, периферийных устройств и данных) между вычислительными процессами, конкурирующими за эти ресурсы

На IBM-совместимые компьютеры устанавливаются операционные системы **Windows** и **Linux**, а на персональные компьютеры **Macintosh** — операционная система **Mac OS**.

Элементы операционной системы

1. Программные модули, управляющие файловой системой.
2. Командный процессор — запрашивает у пользователя команды и выполняет их;
3. Драйверы устройств (обеспечивают управление работой устройств и согласование информационного обмена с другими устройствами, а также позволяют настраивать некоторые параметры устройств).
4. Программные модули, создающие графический интерфейс.
5. Сервисные программы (утилиты) — обслуживают диски, архивируют файлы, обеспечивают работу в сети и т. д.
6. Справочная система (позволяет оперативно получить необходимую информацию о работе самой операционной системы и отдельных ее модулей).

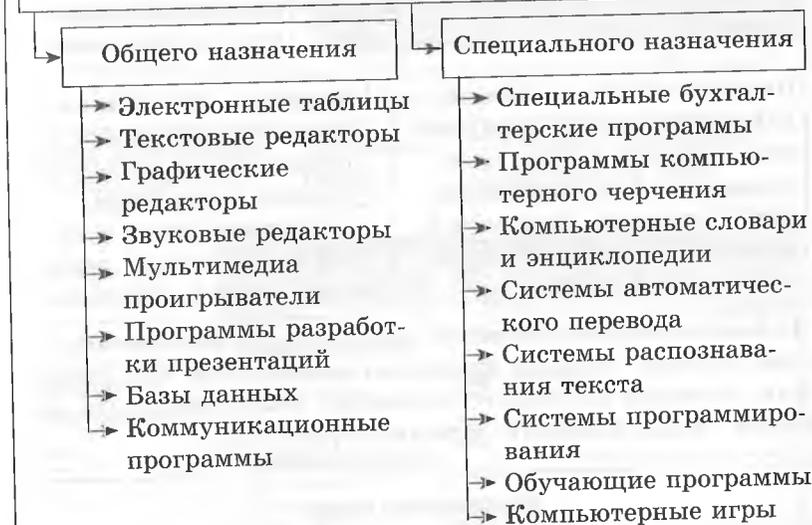
Установка и загрузка операционной системы



Прикладное программное обеспечение

Прикладные программы называются приложениями. Приложение — это программа, позволяющая обрабатывать текстовую, графическую, числовую, аудио- и видеoinформацию, а также работать в компьютерных сетях. Приложение функционирует под управлением операционной системы.

Приложения



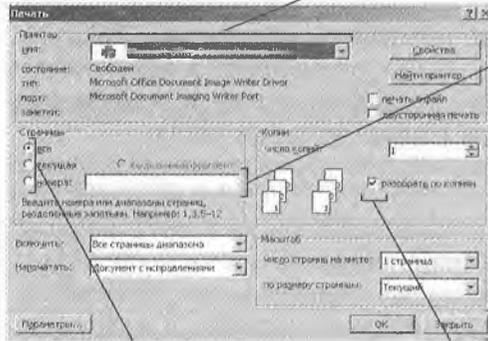
Графический интерфейс операционных систем и приложений

Современные операционные системы для персональных компьютеров обеспечивают взаимодействие с пользователем с помощью графического интерфейса.

Графический интерфейс представляет собой совокупность панелей, меню и окон.

Элементы управления на диалоговой панели

Раскрывающийся список — представляет собой набор значений и выглядит как текстовое поле, снабженное кнопкой с направленной вниз стрелкой.



Текстовое поле — в него можно ввести последовательность символов

Кнопка — обеспечивает выполнение того или иного действия; надпись на кнопке поясняет ее назначение

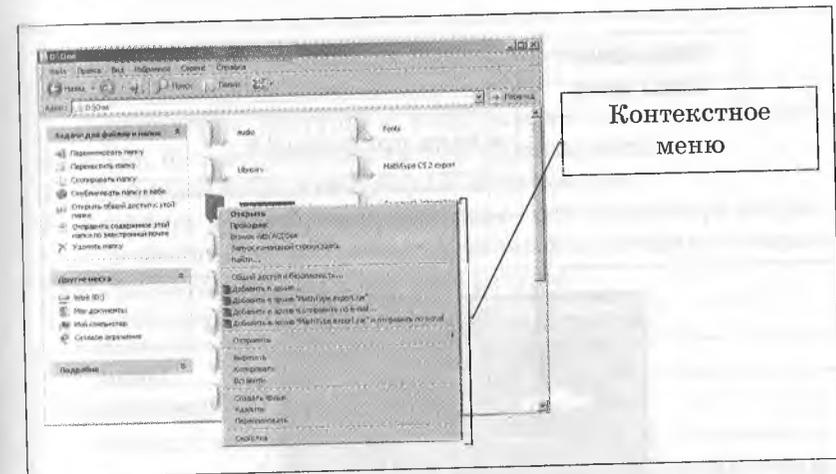
Переключатель — служит для выбора одного из вариантов; варианты выбора представлены в виде маленьких белых кружков. Выбранный вариант обозначается кружком с точкой внутри

Флажок — обеспечивает присвоение параметру определенного значения. Флажок имеет форму квадратика; когда флажок установлен, в нем присутствует «галочка»

На диалоговой панели может также находиться **счетчик** — пара стрелок, которые позволяют увеличивать или уменьшать значение некоторого параметра, и **ползунок**, позволяющий плавно изменять значение параметра.

Контекстное меню

Еще один элемент графического интерфейса — **контекстное меню** объекта, которое вызывается щелчком правой кнопки мыши. Это небольшое окно, которое содержит перечень операций, разрешенных с данным объектом в данном режиме, а также позволяет ознакомиться со свойствами объекта и при необходимости изменить их.



Контекстное меню

Окна

Важнейшим элементом графического интерфейса являются окна.

Основные элементы окна

Меню управления состоянием окна

Заголовок окна

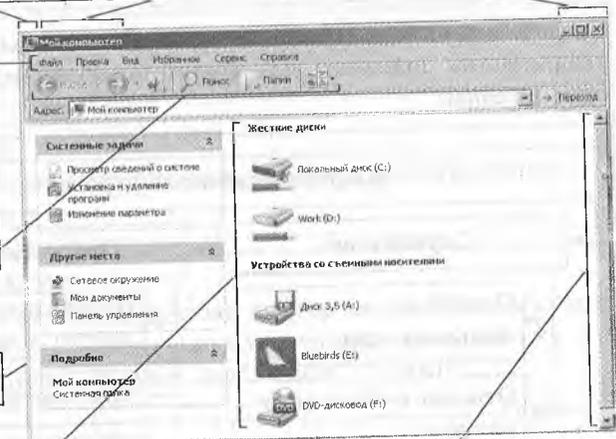
Кнопки управления состоянием окна

Меню окна

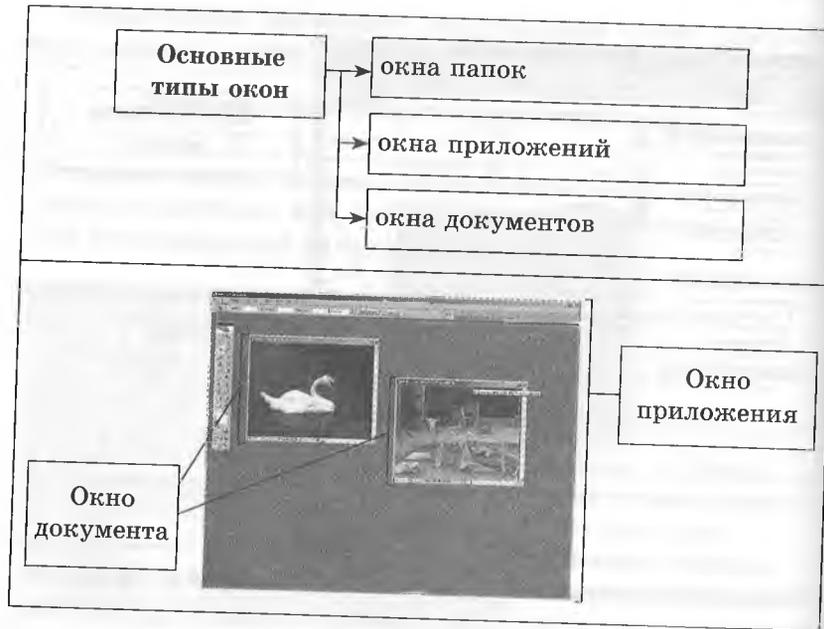
Панели инструментов

Рамка

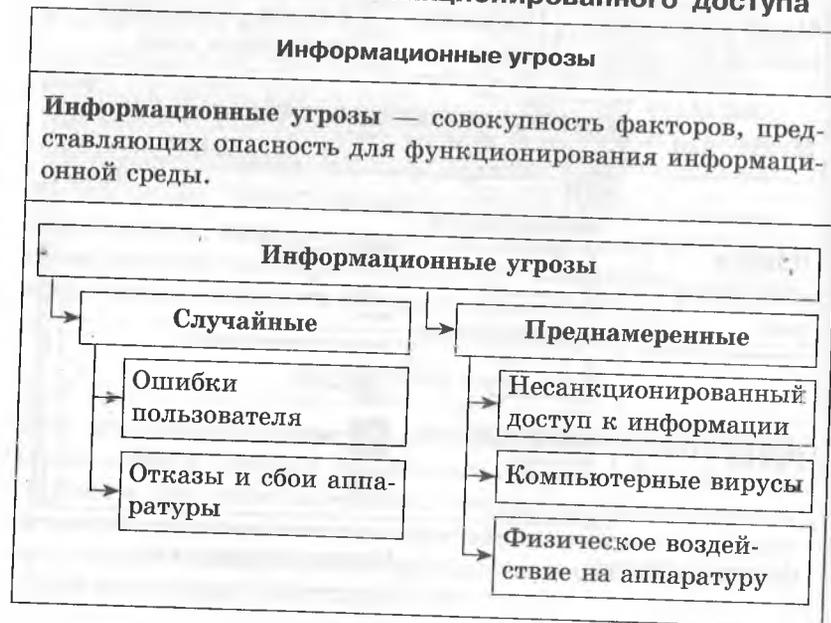
Рабочая область



Полосы прокрутки



Технологии и средства защиты информации от разрушения и несанкционированного доступа



Методы защиты от преднамеренных угроз

1. Ограничение доступа к информации — использование паролей, биометрических систем идентификации.
2. Шифрование (криптография) информации.
3. Контроль доступа к аппаратуре — вся аппаратура закрыта, а в местах доступа к ней установлены датчики, которые срабатывают при вскрытии аппаратуры.
4. Законодательные меры.

Компьютерные вирусы

Компьютерный вирус — это программа, способная создавать свои копии (не обязательно полностью совпадающие с оригиналом), внедрять их в различные объекты или ресурсы компьютерных систем и сетей, а также производить определенные действия без ведома пользователя.

Классификация вирусов

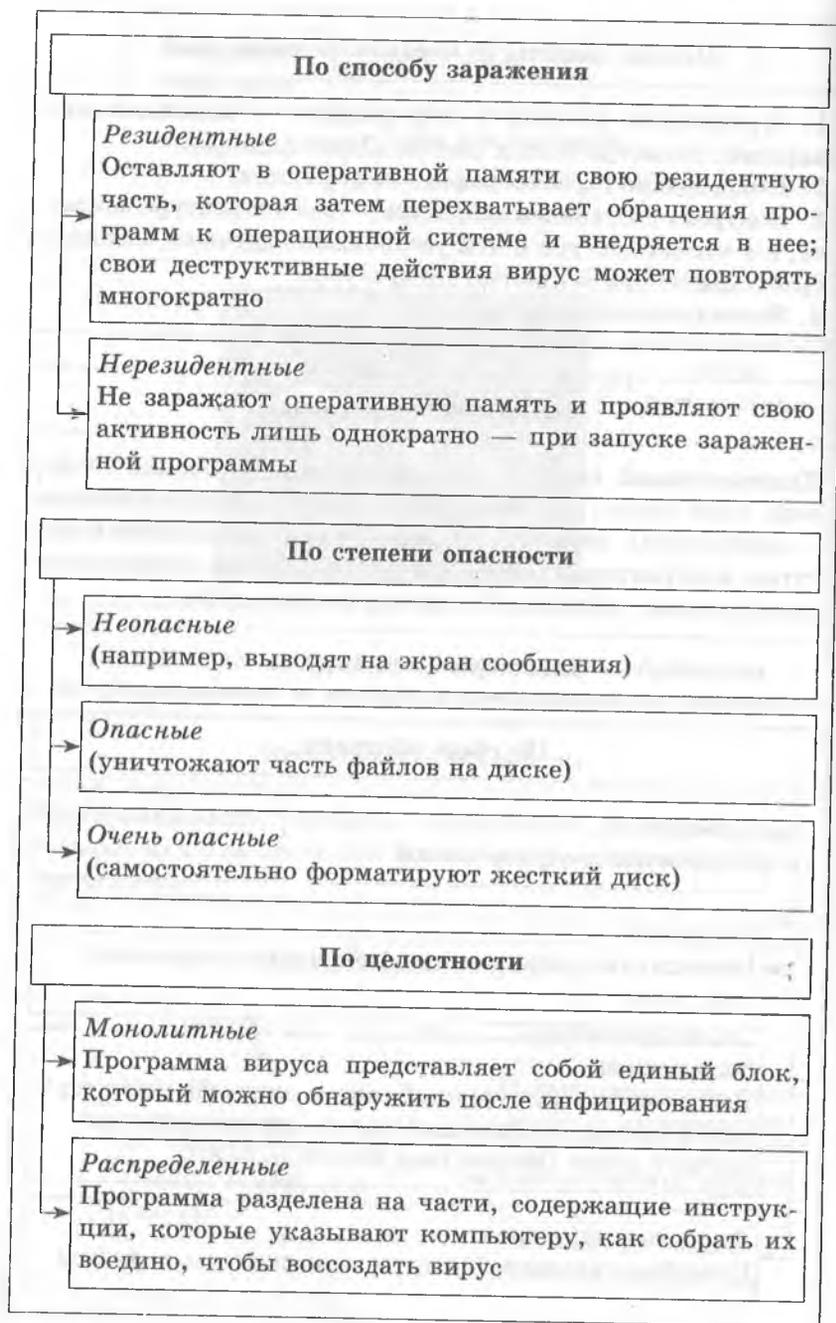
По среде обитания

→ **Сетевые**
Распространяются по сетям

→ **Файловые**
Инфицируют исполняемые файлы с расширениями .exe, .com

→ **Загрузочные**
Внедряются в загрузочный сектор диска (Boot-сектор) или в сектор, содержащий программу загрузки системного диска (Master Boot Record — MBR)

→ **Файлово-загрузочные**
Способны заражать и загрузочные секторы, и файлы



В соответствии с особенностями алгоритма	
1.	Вирусы-компаньоны — создают для ехе-файлов новые файлы-спутники, имеющие то же самое имя, но с расширением com; вирус записывается в com-файл и никак не изменяет одноименный ехе-файл; при запуске такого файла операционная система первым обнаружит и выполнит com-файл, т. е. вирус, который затем запустит и ехе-файл.
2.	Паразитические — изменяют содержимое дисков секторов или файлов.
3.	Репликаторы (черви) — распространяются в сети: проникают в память компьютера, вычисляют сетевые адреса других компьютеров и рассылают по этим адресам свои копии, тем самым уменьшая пропускную способность сети и замедляя работу серверов; могут размножаться без внедрения в другие программы и иметь «начинку» из компьютерных вирусов.
4.	Невидимки (стелс) — маскируют свое присутствие в компьютере; они перехватывают обращения операционной системы к зараженным файлам или секторам дисков и «подставляют» незараженные участки файлов.
5.	Мутанты (призраки, полиморфные вирусы) — их копии практически не содержат полностью совпадающих участков кода; это достигается тем, что в программы вирусов добавляются пустые команды («мусор»), которые не изменяют алгоритм работы вируса, однако затрудняют их выявление.
6.	Макровирусы — используют возможности макроязыков, встроенных в системы обработки данных.
7.	«Троянские кони» — маскируются под полезную или интересную программу, выполняя во время своего действия разрушительную работу (например, стирают FAT и др.).

Антивирусные программы (программы для борьбы с вирусами)

Антивирусные программы

Программы-детекторы (сканеры) — рассчитаны на обнаружение конкретных вирусов. Основаны на сравнении специфической последовательности байтов (сигнатур или масок вирусов), содержащихся в теле вируса, и байтов проверяемых программ. Эти программы нужно регулярно обновлять, так как они быстро устаревают и не могут выявлять новые виды вирусов. Если программа не опознается детектором как зараженная, это не означает, что она «здорова». В ней может быть вирус, который не занесен в базу данных детектора

Программы-доктора (фаги) — не только находят файлы, зараженные вирусом, но и «лечат» их, удаляя из файла тело программы-вируса. Полифаги позволяют «лечить» большое число вирусов. Широко распространены программы-детекторы, одновременно выполняющие и функции программ-докторов, например: AVP (автор Е. Касперский), Aidstest (Д. Лозинский), Doctor Web (И. Данилов), Norton Antivirus (Symantec Corp)

Программы-ревизоры — анализируют текущее состояние файлов и системных областей дисков и сравнивают его с информацией, сохраненной ранее в одном из файлов ревизора. При этом проверяются состояние Boot-сектора (загрузочного сектора), FAT, длина файлов, время их создания, атрибуты, контрольные суммы (суммирование по модулю 2 всех байтов файла). Пример такой программы — Adinf (Д. Мостовой)

Программы-фильтры (сторожа, мониторы) — программы, которые оповещают пользователя обо всех попытках какой-либо программы выполнить подозрительные действия, а пользователь принимает решение о разрешении или запрещении выполнения этих действий. Фильтры контролируют следующие операции: обновление программных файлов и системной области дисков; форматирование диска; резидентное размещение программ в ОЗУ. Примером может служить программа Vsafe. Она не способна обезвредить вирус. Для этого нужно использовать фаги

Программы-иммунизаторы — записывают в вакцинируемую программу признаки конкретного вируса таким образом, что вирус считает ее уже зараженной и поэтому не производит повторное инфицирование. Эти программы морально устарели и менее эффективны

1.5. ОСНОВЫ СОЦИАЛЬНОЙ ИНФОРМАТИКИ

Основные этапы становления информационного общества

Информационные революции — этапы появления средств и методов обработки информации, вызвавших кардинальные изменения в обществе.

Информационные революции

Первая
Изобретение письменности, в результате чего стало возможным фиксирование накопленных знаний и передача их следующим поколениям

<i>Вторая (середина XVI в.)</i>
→ Изобретение книгопечатания, сделавшего культурные ценности общедоступными. Книгопечатание — одна из первых информационных технологий
<i>Третья (конец XIX в.)</i>
→ Начало использования электричества, благодаря чему появились телеграф, телефон и радио, т. е. средства информационной коммуникации
<i>Четвертая (70-е гг. XX в.)</i>
→ Изобретение микропроцессорной технологии и появление персональных компьютеров
<p>Современная цивилизация находится в состоянии перехода от индустриальной фазы своего развития к информационной.</p> <p>Индустриальное общество — общество, определяемое уровнем развития промышленности и ее технической базы.</p> <p>Информационное общество — общество, в котором большинство работающих занято производством, хранением, переработкой, продажей информации и ее обменом.</p>

Некоторые характерные черты информационного общества

1. Решена проблема информационного кризиса, т. е. разрешено противоречие между информационной лавиной и информационным голодом.
2. Обеспечен приоритет информации по сравнению с другими ресурсами.

3. Главной формой развития становится информационная экономика.
4. В основу общества заложены автоматизированные генерация, хранение, обработка и использование знаний с помощью новейшей информационной техники и технологии.
5. Информационные технологии охватывают все сферы социальной деятельности человека.
6. С помощью средств информатики реализован свободный доступ каждого человека к информационным ресурсам всей цивилизации.

Информатизация — процесс перехода от индустриального общества к информационному.
Процесс информатизации общества — базовая составляющая пятой информационной революции.

Хронология создания вычислительных устройств

Устройство	Дата создания
Абак (счетная доска, счеты)	V в. до н.э. — XVIII в. н.э.
Вычислительное устройство Паскаля	1624 г.
Арифмометр Лейбница	конец XVII — начало XVIII в.
Аналитическая машина Беббиджа	1834 г.
Табулятор Холлерита	конец XIX в.
Первая вычислительная машина на электромеханических реле с двоичной кодировкой Z1 Конрада Цузе	1938 г.

Первый компьютер на электронных лампах ABC (Дж. В. Атанасов и К. Берри)	1939 г.
Вычислительная машина на электромагнитных реле Mark-1 (Г. Эйкен)	1939–1944 гг.
Вычислительная машина Colossus-1 с программой, записанной в памяти	Начало 1940-х гг.
Электронная вычислительная машина ENIAC (Дж. Моучли и П. Эккерт)	1943–1946 гг.
Первый компьютер, в котором реализованы принципы фон Неймана — EDSAC	1949 г.
Электронные вычислительные машины различных поколений	от 1940-х гг. до настоящего времени

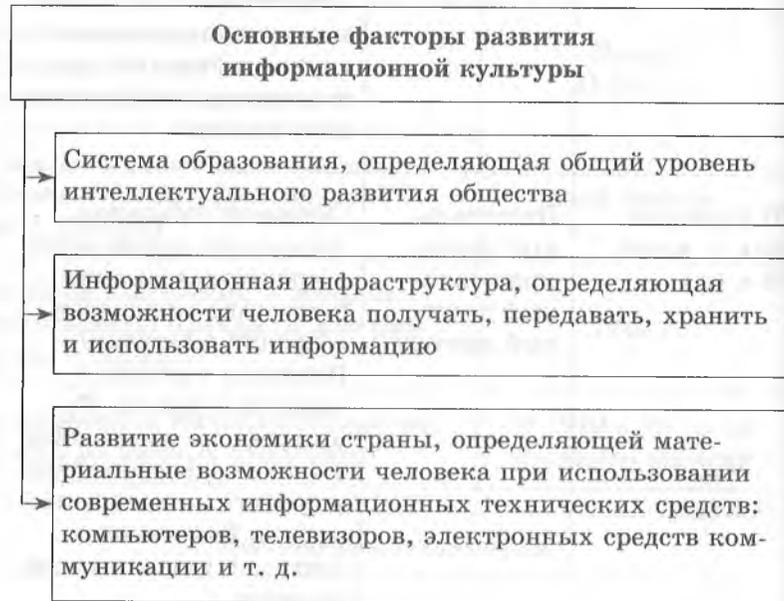
Поколения компьютеров

Поколение	Элементная база	Характеристика
I (середина 40-х — конец 50-х гг.)	Электронные лампы	Большие габариты и значительное потребление энергии, малое быстродействие (до 20 000 операций в секунду), низкая надежность. Программирование в машинных кодах. Использование перфолент и перфокарт для ввода программ и данных

II (конец 50-х — середина 60-х гг.)	Полупроводниковые элементы	Улучшены технические характеристики (быстродействие увеличено до сотен тысяч операций в секунду; емкость внутренней памяти достигла 300 Кбайт). Появились алгоритмические языки и первые операционные системы. Развитие устройств внешней (магнитной) памяти
III (середина 60-х — конец 70-х гг.)	Интегральные схемы, многослойный печатный монтаж	Снижение габаритов, повышение надежности и производительности (несколько миллионов операций в секунду). Появился мультипрограммный режим. Использование нового типа внешних запоминающих устройств — магнитных дисков. Возможность доступа с удаленных терминалов
IV (конец 70-х гг. — настоящее время)	Микропроцессоры, большие интегральные схемы	Улучшенные технические характеристики. Масовый выпуск персональных компьютеров
V (перспективы развития)	Машины, основанные на искусственном интеллекте	

Информационная культура

Информационная культура — важнейшая составляющая культуры в целом. Это — умение целенаправленно работать с информацией и использовать для ее получения, обработки и передачи компьютерную информационную технологию, современные технические средства и методы.



Этические и правовые нормы информационной деятельности людей

Информационная культура предполагает знание и соблюдение этических норм и правил. В большинстве своем информационные этические нормы совпадают с общечеловеческими нравственными принципами. Это **порядочность, честность, точность, корректность, объективность** в оценке и представлении информации. Всегда необходимо иметь в виду **степень конфиденциальности информации**, с которой приходится работать, не допускать использования информации неподобающим образом.

Этические нормы в области информатики

При работе в сети следует руководствоваться определенными общеизвестными нормами этики общения.

Общение в чатах	Общение по электронной почте	Общение в телеконференциях
В своих сообщениях следует использовать язык, на котором общается большинство присутствующих; здороваться при появлении в чате; придерживаться темы разговора, если чат специализируется на определенной тематике; не использовать ненормативную лексику	В частной переписке стиль письма может быть любым, если он приемлем адресатом. В деловой переписке нужно изъясняться кратко и грамотно; подписывать письма; заполнять поле «Тема» (чтобы сообщение не было принято за спам или вирус и удалено без прочтения); отвечать на письма вовремя; не пересылать большие файлы без предварительной архивации; не рассылать знакомым и незнакомым людям рекламные сообщения	Сообщения должны быть краткими и по существу обсуждаемой проблемы; не допускается самореклама; сообщения должны быть адресованы всем собеседникам (для частных писем служит электронная почта); запрещены высказывания расистского характера, оскорбления и некорректные замечания
При общении в сети необходимо придерживаться здравого смысла, правил хорошего тона и следовать старой мудрости: «Поступай с другими так, как хочешь, чтобы поступали с тобой».		

Правовые нормы в области информатики

Информация является объектом собственности так же, как и материальные блага (машина, дом, мебель и др.).

Компоненты права собственности

→ *Право распоряжения* состоит в том, что только владелец информации имеет право определять, кому эта информация может быть предоставлена

→ *Право владения* обеспечивает владельцу информации ее хранение в неизменном виде. Никто, кроме самого владельца, не может ее изменять

→ *Право пользования* предоставляет владельцу информации право ее использования только в своих интересах

Любой пользователь обязан приобрести права, прежде чем воспользоваться интересующим его информационным продуктом!

Закон Российской Федерации «Об информации, информатизации и защите информации» (1995 г.) — базовый юридический документ, обеспечивающий цивилизованные правовые отношения на информационном рынке.

РАЗДЕЛ 2. ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

2.1. ТЕХНОЛОГИЯ СОЗДАНИЯ И ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

Текстовые редакторы и документы

Текстовый редактор — это программа, предназначенная для обработки текстовой информации.

Текст, оформленный с помощью текстового редактора, называется **текстовым документом**. Текстовым документом называется также файл, в котором этот текст хранится.

Текстовый процессор — текстовый редактор с расширенным набором функций.

Основные функции текстового редактора

1. Создание текстового документа.
2. Редактирование и форматирование документов.
3. Создание и форматирование таблиц.
4. Внедрение в документ рисунков, формул, диаграмм и т. д.
5. Сохранение документов.
6. Печать документов.

Интерфейс текстового редактора Microsoft Word 2007

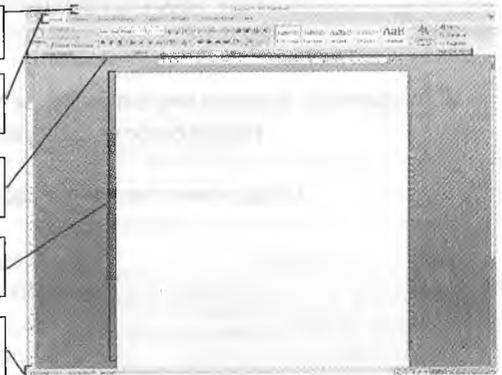
Строка меню

Строка заголовка

Панель инструментов

Рабочий лист

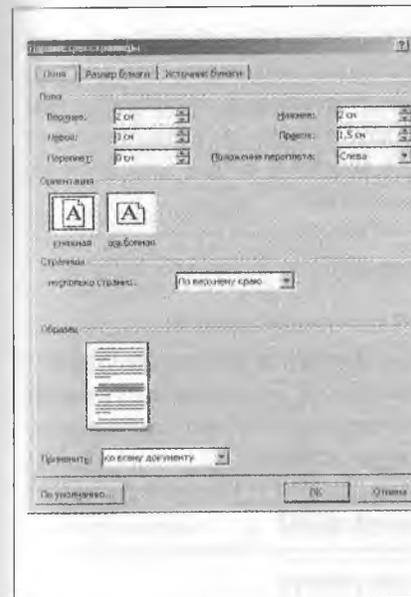
Строка состояния



Назначение интерфейсных элементов программы	
Элемент	Назначение
Строка заголовка	Содержит стандартные кнопки для работы с окном и название текущего документа
Строка меню	Включает в себя все команды для работы с текстовым документом
Панель инструментов	Содержит кнопки, которые наиболее часто используются при работе с документом. Кнопки панели инструментов дублируют пункты меню. Включать-выключать панели можно с помощью команд из группы <i>Панель инструментов</i> на вкладке <i>Вид</i>
Рабочее поле	Предназначено для ввода и редактирования текста документа. Рабочее поле содержит также горизонтальную и вертикальную линейки и полосу прокрутки
Строка состояния	Содержит информацию о текущем состоянии документа: номер текущей страницы, номер раздела, общее количество страниц в документе, расстояние текущего положения курсора от верхнего края страницы в сантиметрах, номер строки, номер столбца, индикаторы режимов работы программы

Создание, редактирование и форматирование текстового документа

Создание текстового документа	
	<p>Текстовый документ разбит на страницы, поэтому начинать работу над документом целесообразно с задания параметров страницы: формата, ориентации, размера полей и т. д. Все это выполняется с помощью команд из группы <i>Параметры страницы</i> на вкладке <i>Разметка страницы</i>.</p> <p>Задание параметров страницы можно также выполнить с помощью диалогового окна <i>Параметры страницы</i>. Окно открывается при щелчке на кнопке с изображением стрелки рядом с названием группы.</p>

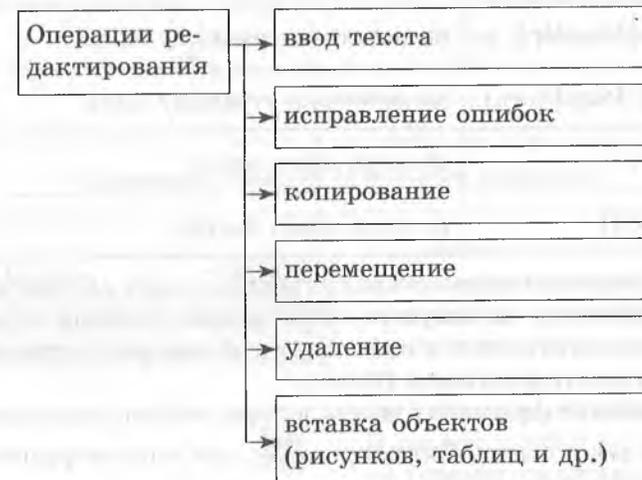


но с задания параметров страницы: формата, ориентации, размера полей и т. д. Все это выполняется с помощью команд из группы *Параметры страницы* на вкладке *Разметка страницы*.

Задание параметров страницы можно также выполнить с помощью диалогового окна *Параметры страницы*. Окно открывается при щелчке на кнопке с изображением стрелки рядом с названием группы.

Редактирование текстового документа

Редактирование — это изменение содержания документа.



Ввод текста происходит с клавиатуры. В рабочей области располагается *курсор* — мигающая вертикальная черта. Символы появляются в месте расположения курсора, а сам он сдвигается вправо. Установить курсор в нужное место текста можно щелчком мыши в нужной точке или *курсорными клавишами* и их комбинациями:

[Home]	перемещает курсор в начало текущей строки
[End]	перемещает курсор в конец текущей строки
[Ctrl]-[→]	на одно слово вперед
[Ctrl]-[←]	на одно слово назад
[Ctrl]-[Home]	в начало документа
[Ctrl]-[End]	в конец документа
[PageUp]	на экранную страницу вверх
[PageDown]	на экранную страницу вниз
[Ctrl]-[PageUp]	на печатную страницу вверх
[Ctrl]-[PageDown]	на печатную страницу вниз
[Ctrl]-[↑]	на один абзац назад
[Ctrl]-[↓]	на один абзац вперед

По достижении правого края страницы текст автоматически переносится на новую строку. Процесс принудительного завершения строки и начала нового абзаца осуществляется путем нажатия клавиши [Enter].

Копирование фрагмента текста в буфер обмена выполняется с помощью команды *Копировать* ; для вставки фрагмента

из буфера обмена нужно установить курсор в требуемое место, а затем использовать команду *Вставить* . **Перемещение** фрагмента можно выполнить перетаскиванием мыши или с помощью команд *Вырезать*  и *Вставить* . **Удаление** текста или его фрагментов производится нажатием клавиши [Del].

Форматирование текстового документа

Форматирование — это изменение внешнего вида документа и его частей.

Операции форматирования

→ изменение свойств отдельных символов

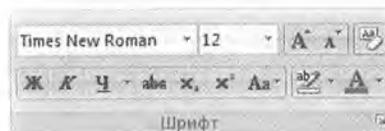
→ изменение параметров абзацев

→ оформление списков

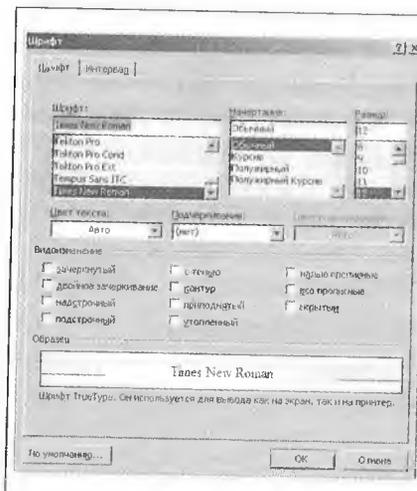
→ оформление заголовков

→ вставка колонтитулов и номеров страниц и др.

Изменение свойств отдельных символов

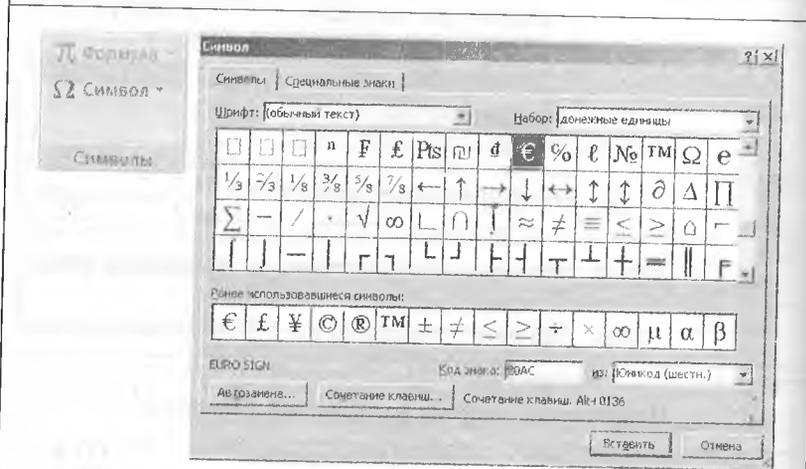


По умолчанию все операции изменения шрифта применимы к выделенному фрагменту текста, а если такого нет — к слову, на котором стоит курсор. В группе *Шрифт* на вкладке *Главная* представлены



команды настройки гарнитуры, начертания, цвета, размера и дополнительных эффектов шрифта. Задание параметров шрифта также можно выполнить с помощью диалогового окна **Шрифт**. Окно открывается при щелчке по кнопке с изображением стрелки рядом с названием группы.

Вставка символов



В текстовых документах можно встретить символы, которых нет на клавишах клавиатуры. Эти символы вставляются с помощью группы **Символ** на вкладке **Вставка**. Их можно также вставить с помощью диалогового окна **Символ**.

Абзац и его свойства

Абзац — фрагмент текста, начинающийся с новой строки и заканчивающийся непечатаемым символом «¶».



Параметры абзацев можно задать с помощью команд из группы **Абзац** вкладки **Главная**, а также с помощью диалогового окна **Абзац**. Оно открывается при щелчке по кнопке с изображением стрелки рядом с названием группы. Диалоговое окно **Абзац**, содержит две вкладки: **Отступы и интервалы** и **Положение на странице**. На вкладке **Отступы и интервалы** можно задать способ выравнивания текста, правую и левую границы абзаца относительно границ страницы, «красную строку», интервалы между абзацами и между строками данного абзаца. Вкладка **Положение на странице**

используется для абзацев, попадающих на границу между страницами. Можно запретить отрывать от абзаца одну строку, потребовать, чтобы абзац размещался на одной странице целиком, начать текущим абзацем новую страницу.

Форматировать абзац можно также с помощью линейки форматирования. Перетаскиванием по линейке треугольного маркера слева внизу (¶) задают левую границу абзаца, справа внизу (¶) — правую границу, слева вверх (¶) — «красную строку».

Оформление списков

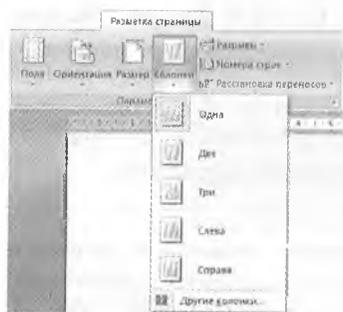
Упорядоченную информацию часто удобнее представлять в виде списков (инструкции, перечни предметов или объектов и т. д.). Программа Word поддерживает два вида списков: *маркированные*, в которых каждый пункт помечается маркером, и *нумерованные*, где пункты последовательно нумеруются.



Для преобразования существующего текста в нумерованный или маркированный список нужно выделить этот текст и выбрать команду *Нумерация* или *Маркеры* из группы

Абзац на вкладке *Главная*. Команда *Многоуровневый список* позволяет задать специальный список, содержащий до девяти уровней пунктов, которые нумеруются или маркируются отдельно. Для перехода на более низкий уровень служит команда *Увеличить отступ* из группы *Абзац*. Для возврата на более высокий уровень используется команда *Уменьшить отступ*.

Оформление текстовых колонок



В газетах и некоторых книгах (например, в словарях и энциклопедиях) информацию для облегчения восприятия часто представляют в несколько столбцов. Для разбивки предварительно выделенного текста на колонки можно воспользоваться командой *Колонки* из группы *Параметры страницы*

на вкладке *Разметка страницы*. В открывшемся меню можно выбрать один из предлагаемых способов разбивки текста на колонки или воспользоваться пунктом *Другие колонки* для их более полной настройки.

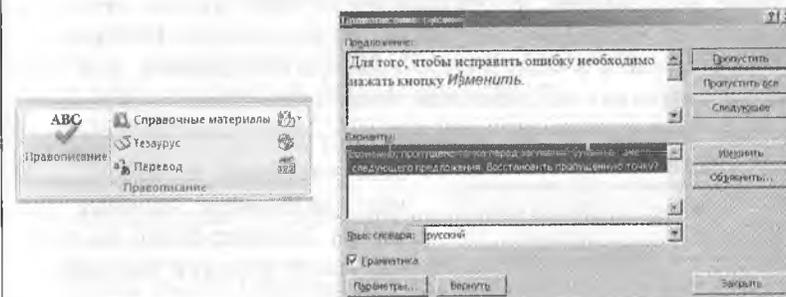
Автоматизация обработки документов

Текстовый редактор Microsoft Word является мощной автоматизированной системой, в которой предусмотрены инструменты автоматизации обработки текстовых документов.

Инструменты автоматизации работы с текстовыми документами

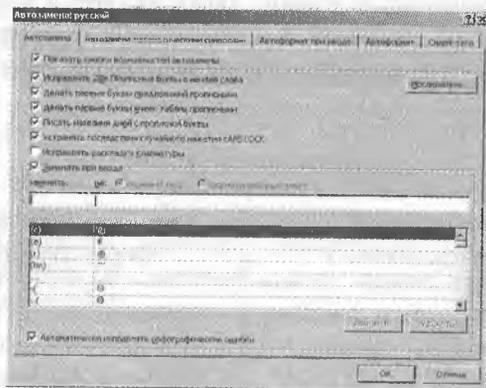
Автоматизация редактирования	Автоматизация форматирования
<ul style="list-style-type: none"> • проверка орфографии • автозамена • автотекст • поиск и замена символов 	<ul style="list-style-type: none"> • автоперенос • нумерация страниц • стилевое форматирование • создание оглавления • перекрестные ссылки • автонумерация таблиц и рисунков

Проверка орфографии



Для проверки орфографии необходимо установить курсор в то место, откуда необходимо начать проверку, и воспользоваться командой *Правописание* на вкладке *Рецензирование*. Система автоматически начнет проверку. При обнаружении ошибки откроется диалоговое окно, в котором будут предложены варианты исправления ошибки. Для того чтобы исправить ошибку, необходимо нажать кнопку *Изменить*.

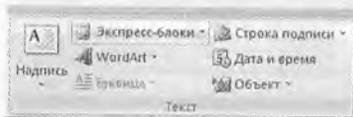
Автозамена и автотекст



Для автоматизации ввода и исправления текста в среде Word используются инструменты *Автозамена* и *Автотекст*.

Для включения параметра *Автозамена* нужно нажать кнопку *Microsoft Office* , затем кнопку *Параметры Word*, после

чего выбрать пункт *Правописание* и нажать кнопку *Параметры автозамены*. В открывшемся окне нужно отметить флажками желаемые настройки. Нажав кнопку *Исключения*, можно перейти в окно *Исключения при автозамене*, для того чтобы запретить исправление некоторых слов.

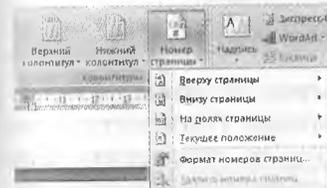


Для создания и использования автотекста служит команда *Экспресс-блоки* из группы *Текст* вкладки *Вставка*.

Поиск и замена

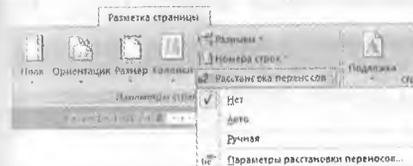
Для автоматического поиска и замены символов в среде Word используется инструмент поиска и замены символов — команды *Найти* и *Заменить* из группы *Редактирование* вкладки *Главная*.

Нумерация страниц



Пронумеровать страницы в документе и выполнить форматирование номеров можно, воспользовавшись командой *Номер страницы* из группы *Колонтитулы* вкладки *Вставка*.

Автоматическая расстановка переносов



Для автоматической расстановки переносов в тексте необходимо использовать команду *Расстановка переносов* из группы *Параметры страницы* вкладки *Разметка страницы*.

Стилевое форматирование

Стиль — это заранее определенная и сохраненная совокупность всех параметров оформления текста.

Виды стилей

→ стиль *абзаца* (определяет внешний вид абзаца)

→ стиль *знака* (определяет внешний вид того или иного символа)

→ стиль *списка* (устанавливает для выделенных абзацев одинаковые параметры списка)

→ стиль *таблицы* (устанавливает значения параметров таблицы)



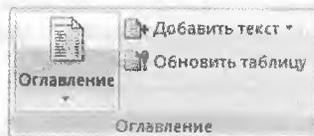
Доступные стили указаны в списке группы *Стили* на вкладке *Главная*. В начале работы с программой Word этот список содержит перечень стилей, заданных по умолчанию. При выборе одного из стилей, представленных в списке, изменяется формат текущего абзаца (в котором расположен курсор) или формат выделенного фрагмента. Изменить стиль можно также с помощью команды *Изменить стили* из группы *Стили*.

Можно создать свой собственный стиль форматирования. Для этого нужно открыть окно *Стили* щелчком на кнопке с изображением стрелки рядом с названием группы *Стили* и в нем щелкнуть по кнопке *Создать стиль*.

Создание оглавления и предметного указателя



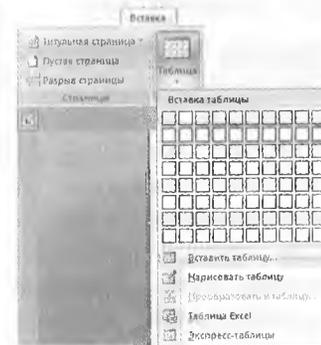
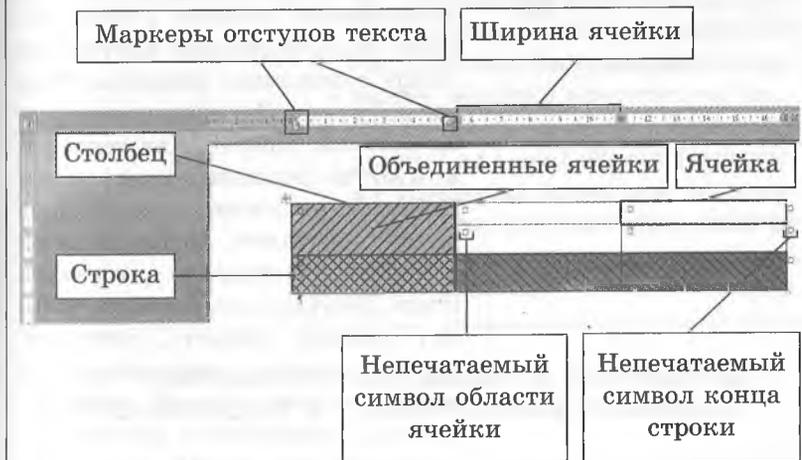
Для создания оглавления необходимо воспользоваться командой из группы *Оглавление* вкладки *Ссылки*.



Для создания предметного указателя необходимо воспользоваться командой из группы *Предметный указатель* вкладки *Ссылки*.

Создание и редактирование таблиц

Устройство электронной таблицы



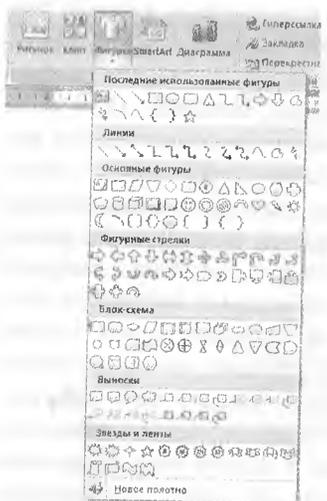
Вставить таблицу в документ можно с помощью команд *Нарисовать таблицу* или *Вставить таблицу* из группы *Таблицы* вкладки *Вставка*. При этом открывается дополнительное меню *Работа с таблицами*, содержащее вкладки *Конструктор* и *Макет*. С помощью вкладки *Конструктор* можно выбрать стиль таблицы и настроить границы и заливку ячеек таблицы. С помощью вкладки *Макет*

можно изменить структуру таблицы, удалив или добавив строки, столбцы или отдельные ячейки, а также изменить размеры ячеек, задать направление и способ выравнивания текста в ячейках, объединить несколько ячеек или разбить их, установить поля ячеек и интервалы между ячейками. Команды из группы *Данные* вкладки *Макет* позволяют также выполнить несложную обработку содержимого таблицы.

Создание и редактирование графических объектов



Программа Word предоставляет возможность вставить в текст заранее подготовленный рисунок из файла с помощью команды *Рисунок* из группы *Иллюстрации* вкладки *Вставка*. При этом на ленте открывается дополнительное меню *Работа с рисунками*, содержащее вкладку *Формат*. Команды этой вкладки позволяют изменить стиль рисунка, его размеры, форму, границу, яркость, контрастность, задать дополнительные эффекты (тень, рельеф, свечение и т. д.), переместить рисунок на передний или задний план, указать способ обтекания текстом.



Простые рисунки можно выполнить, используя средства самой программы WORD. Команда *Фигуры* из группы *Иллюстрации* вкладки *Вставка* позволяет создать одну из стандартных фигур. При этом на ленте открывается дополнительное меню *Средства рисования*, содержащее вкладку *Формат*. Команды этой вкладки служат для изменения стиля фигуры, ее размеров, контура, заливки, эффектов тени и объема, для добавления текста, группирования нескольких фигур в единый рисунок, задания положения рисунков в тексте, их перемещения на передний или задний план.



Команда *Диаграмма* из группы *Иллюстрации* вкладки *Вставка* позволяет разместить в документе диаграмму. При этом на ленте открывается дополнительное меню *Работа с диаграммами*, содержащее вкладки *Конструктор* (выбор стиля диаграммы и исходных данных для нее), *Макет* (настройка отдельных элементов диаграммы) и *Формат* (изменение размера диаграммы, задание дополнительных эффектов).

2.2. ТЕХНОЛОГИЯ СОЗДАНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ В ЭЛЕКТРОННЫХ ТАБЛИЦАХ

Ввод и редактирование данных в электронных таблицах, операции с данными

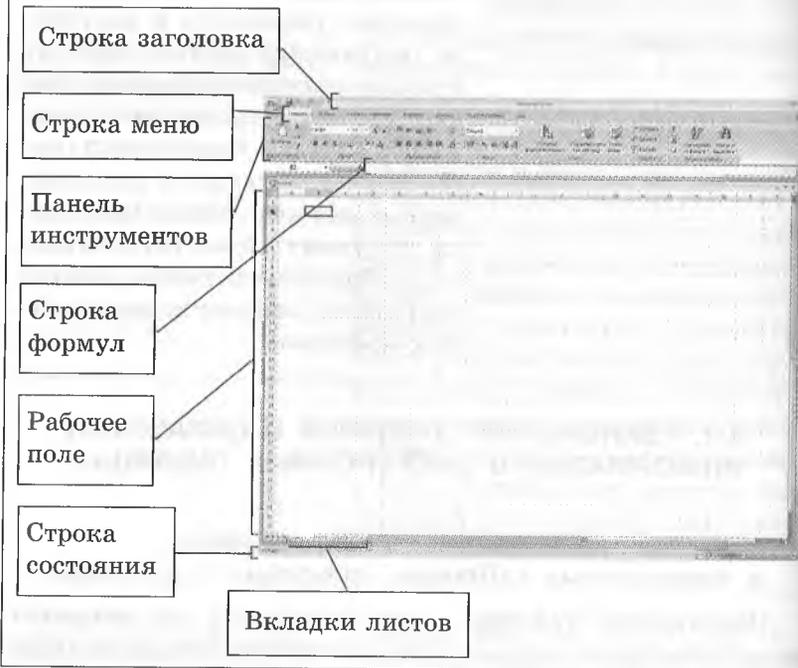
Электронная таблица — это программа для автоматической обработки данных, представленных в виде таблицы.

Программа Microsoft Excel — одно из самых популярных средств управления электронными таблицами.

Основные функции электронных таблиц

1. Организация ввода данных в таблицу и сохранение их в памяти компьютера.
2. Редактирование данных таблицы.
3. Форматирование данных таблицы.
4. Иллюстрирование таблиц (рисунки, анимации и т. д.).
5. Обработка табличных данных.
6. Упорядочение и сортировка данных.
7. Использование макросов для выполнения вычислений по данным таблицы.
8. Построение диаграмм и графиков по данным таблицы.
9. Печать таблиц.

Интерфейс электронных таблиц Microsoft Excel 2007



Основные объекты электронных таблиц

Основные объекты электронных таблиц — ячейка, таблица, лист и книга.

Любой документ, созданный в Excel, называется книгой. Каждая рабочая книга хранится в отдельном файле с расширением `.xls` или `.xlsx`. Книга состоит из совокупности таблиц, называемых листами. Каждый лист имеет собственное уникальное имя. Лист представляет собой прямоугольную таблицу, состоящую из 256 столбцов и 65 535 строк. **Столбцы** обозначаются латинскими буквами *A, B, C, D* и т. д. Если букв алфавита не хватает — используют двухбуквенные обозначения *AA, AB, AC, ..., IV*.

Ячейка — минимальный элемент электронной таблицы.

Виды данных, которые могут содержать ячейки

Текстовые данные

Строка текста произвольной длины

Числовые данные

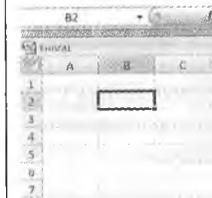
Отдельное число (в допустимом формате). Как числа рассматриваются также данные, определяющие даты или денежные суммы

Формулы

Содержимое ячейки рассматривается как формула, если оно начинается со знака равенства «=». Ячейка, содержащая формулу, является вычисляемой, т. е. значение ее может зависеть от значений других ячеек таблицы. Формулы в ячейках не отображаются. Вместо формулы воспроизводится полученный при ее вычислении результат

Таблицы могут также содержать рисунки, диаграммы, анимационные ролики и другие объекты. Все эти объекты размещаются не в ячейках таблицы, а независимо от них, занимая произвольную часть листа.

Адрес ячейки в электронной таблице



Ячейки располагаются на пересечении столбцов и строк. Номер (адрес) ячейки формируется из номера столбца и номера строки без пробела между ними. Например, *A1, D24, M355*. Одна из ячеек на рабочем листе всегда является активной (текущей). Она обведена широкой рамкой, ее номер выводится в поле имени, а в строке формул размещается ее содержимое. Именно с этой ячейкой производится работа — считывание или запись.

Ввод данных

Данные вводятся в текущую ячейку и одновременно отображаются в строке формул. Вводимый текст автоматически выравнивается по левому краю, а числа — по правому. Необходимые данные можно вставить в электронную таблицу из других приложений с помощью команд группы *Получить внешние данные* вкладки *Данные* или путем обычного копирования с использованием буфера обмена Windows.

Редактирование введенных данных

Если нужно ввести новые данные в заполненную ячейку, необходимо сделать ее текущей и начать вводить новые значения. Старое содержимое ячейки при этом удаляется. Если же нужно лишь откорректировать данные, то следует, активизировав ячейку, нажать клавишу [F2] или выполнить двойной щелчок левой кнопкой мыши по ячейке. При этом в ячейке появится текстовый курсор, и содержимое можно исправить. Исправления также можно ввести в строку формул активной ячейки.

Автоматизация ввода данных

В электронных таблицах реализована автоматизация ввода данных. Многие таблицы могут содержать в ячейках повторяющиеся данные. При заполнении таких таблиц Excel уже по первой букве «догадывается», что хочет ввести пользователь, и выводит подсвеченное значение в ячейку.

Чтобы **размножить** данные на несколько ячеек, нужно сделать активной первую ячейку диапазона и заполнить ее. Затем установить указатель мыши на *маркер заполнения* (правый нижний угол рамки текущей ячейки).

Перетаскиванием маркера заполнения можно *размножить* содержимое текущей ячейки на несколько ячеек в столбце или строке. Если в первой ячейке были введены день недели или дата, то при перетаскивании вправо или вниз их значения будут увеличиваться на 1, а при перетаскивании влево или вверх — уменьшаться на 1.

Если в ячейку было введено число (или текст), то при перетаскивании маркера оно скопируется в другие ячейки. Для создания **числового ряда** при перетаскивании маркера заполнения следует удерживать нажатой клавишу [Ctrl]. По ходу перетаскивания содержимое последней ячейки отображается в небольшом всплывающем окне. Если для перетаскивания маркера использовать правую кнопку мыши, то при ее отпускании появится контекстное меню, позволяющее изменить способ заполнения диапазона.



Если требуется более сложный закон изменения последовательности значений, то после выбора первой ячейки и ввода в нее нужного значения следует выполнить команду *Заполнить/Прогрессия* из группы *Редактирование* вкладки *Главная*.

Операции с данными

Удаление



При нажатии клавиши [Del] произойдет очистка содержимого ячейки. Реальное удаление ячейки сопровождается изменением структуры таблицы. По команде *Удалить ячейки* из группы *Ячейки* вкладки *Главная* откроется диалоговое окно *Удаление ячеек*, в котором нужно выбрать направление смещения ячеек, занимающих освобождающееся место. Аналогичное окно открывается и при выборе команды *Удалить* из контекстного меню ячейки.

Для удаления строки (столбца) нужно выделить какую-либо ячейку в удаляемой строке (столбце) и выполнить команду *Удалить строки с листа* (*Удалить столбцы с листа*) из группы *Ячейки* вкладки *Главная*. Удалить строку (столбец) целиком можно также, выделив строку (столбец) щелчком мыши по маркеру строки (столбца) и выбрав команду *Удалить* из контекстного меню.

Способы копирования и перемещения

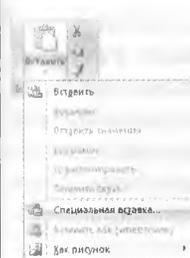
Навести указатель мыши на границу текущей ячейки (диапазона) и перетащить маркер при нажатой левой кнопке мыши на новое место. В результате произойдет **перемещение** ячейки (диапазона). Если при перетаскивании удерживать нажатой клавишу [Ctrl], то произойдет **копирование** ячейки. При перетаскивании появляется серая рамка, которая помогает правильно разместить блок на новом месте.

Выделить ячейку (диапазон), перейти на вкладку *Главная* и выполнить команды *Копировать*  или *Вырезать*  из группы *Буфер обмена*, а затем щелкнуть в нужном месте и выполнить команду *Вставить* .

Выполнить специальное перетаскивание нужного блока ячеек (при нажатой правой кнопке мыши). При отпускании кнопки мыши будет выведено контекстное меню, содержащее перечень допустимых операций: копирование, перемещение, копирование только значений или только форматов, копирование или перемещение со сдвигом остальных ячеек в нужную сторону.

Вставка

Для вставки в таблицу столбца нужно щелкнуть по заголовку столбца, перед которым следует вставить новый столбец, чтобы выделить его. Затем выполнить команду *Вставить* из контекстного меню выделенного столбца. Аналогичным образом можно вставить строку, ячейку или блок ячеек.

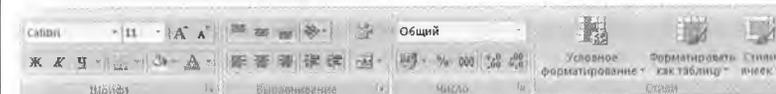


Если нужно скопировать только часть информации, можно воспользоваться специальной вставкой. Команда *Вставить/Специальная вставка* из группы *Буфер обмена* откроет диалоговое окно, в котором можно указать, какую часть информации следует вставить из буфера обмена — например, только значения, без формул, по которым они получены; только форматы или только примечания.



Для вставки в таблицу рисунков или видеоклипов следует выбрать соответствующие параметры из группы *Иллюстрации* вкладки *Вставка*. Простой рисунок можно сделать и самому, воспользовавшись параметрами *Фигуры* и *SmartArt* из группы *Иллюстрации*. При добавлении фигуры на лист, открывается дополнительная вкладка *Средства рисования*, параметры которой позволяют изменить размеры и стиль фигуры, задать дополнительные эффекты, упорядочить, сгруппировать несколько фигур, добавить надписи и т. д.

Форматирование ячеек



Отформатировать ячейку (выделенный диапазон ячеек) и ее содержимое можно с помощью параметров групп *Шрифт*, *Выравнивание*, *Число* и *Стили* вкладки *Главная*. Щелчок по кнопке с изображением стрелки рядом с названием группы открывает диалоговое окно *Формат ячеек*. На вкладке *Число* этого окна можно выбрать формат представления чисел и задать различные параметры, зависящие от выбранного

формата; на вкладке *Выравнивание* установить способ вертикального и горизонтального выравнивания данных в ячейке, задать перенос по слогам, объединение ячеек, автоподбор их ширины и т. д.; с помощью команд вкладки *Шрифт* задать гарнитуру шрифта и его параметры; на вкладках *Граница* и *Вид* задать способ обрамления ячейки, выбрать цвет фона и узор заполнения; с помощью команд вкладки *Защита* защитить ячейку паролем.

Типы и формат данных. Работа с формулами. Абсолютная и относительная ссылки. Использование функций

Три основных типа данных, используемых при работе с электронными таблицами

Число

Текст

Формула

Числа в Excel отображаются в таких категориях форматов, как *Числовой*, *Экспоненциальный*, *Финансовый*, *Денежный*, *Процентный* и *Дробный*. Если необходимо вводить числа без привязки к каким-либо форматам, то в этом случае можно использовать формат *Общий*. При использовании числового формата можно задавать количество десятичных знаков с помощью кнопок *Формат с разделителями* $\frac{000}{000}$, *Увеличить разрядность* $\frac{+00}{+00}$, *Уменьшить разрядность* $\frac{-00}{-00}$ в группе *Число*. Перевести число в процентный формат позволяет кнопка $\%$, а в денежный — кнопка $\$$.
Формат *Дата и время* позволяет хранить значения временных данных в удобном и привычном виде.

Текстом в Excel считается любая последовательность, состоящая из цифр, пробелов и нецифровых символов.

Формула в Excel должна начинаться со знака равенства и может включать в себя числа, имена ячеек, функции и знаки математических операций:

«+» — сложение;

«-» — вычитание;

«*» — умножение;

«/» — деление;

«^» — возведение в степень.

Для того чтобы в формуле сделать ссылку на ячейку, нужно указать ее имя вручную или просто щелкнуть по этой ячейке мышью. При этом ячейка обводится цветной рамкой. Если содержимое ячейки, используемое в формуле, изменится, то и значение формулы изменится.

Типы ссылок на адреса ячеек

Относительная

Указание на ячейку основывается на ее положении относительно той ячейки, в которой находится формула, например: «на две строки выше».

Относительная ссылка на ячейку A1 имеет вид = A1

Абсолютная

Для указания на ячейку используется ее фиксированное положение на листе, например: «ячейка находится в столбце A строки 2».

Абсолютная ссылка на ячейку A1 имеет вид = \$A\$1

Смешанная

Содержит относительную и абсолютную ссылки, например: «ячейка находится в столбце A и выше на две строки».

Комбинируя абсолютные и относительные ссылки на ячейку A1, можно создать следующие смешанные ссылки: = \$A1; = A\$1

Формулы с текстовыми значениями

В большинстве случаев работа с текстовыми значениями происходит так же, как с числами. Например, если ячейка *A1* содержит текст *ABCDE* и в ячейку *A10* введена формула = *A1*, то в ячейке *A10* также будет выведено *ABCDE*.

Для объединения текстовых значений используется текстовый оператор &. Например, если ячейка *A1* содержит текст *ABCDE*, а ячейка *A2* — текст *WQRT* и в ячейку *A3* введена формула = *A1 & A2*, то в ячейке *A3* будет выведено *ABCDEWQRT*. Чтобы вставить пробел между двумя строками, нужно изменить формулу, записав: = *A1&" "* *&A2*. Эта формула использует два текстовых оператора и текстовую константу — в данном случае пробел, заключенный в двойные кавычки.

С помощью оператора & можно объединять и числовые значения. Например, если ячейка *A1* содержит числовое значение *123* и в ячейке *A2* записано число *456*, то результатом формулы = *A1&A2* будет строка символов *123456*. Эта строка выравнивается по левому краю, так как она является текстовым значением.

Оператор & можно использовать для объединения текстовых и числовых значений. Например, если ячейка *A1* содержит текст *ABCDE*, а ячейка *A2* — числовое значение *123*, то результатом формулы = *A1&A2* будет строка *ABCDE123*.

Если Excel не может вычислить результат формулы, то в ячейку выводится *ошибка*, например: #*Дел/0!*, #*ИМЯ?*, #*ЗНАЧ!* и др.

Пример 1. В ячейке *C1* записана формула = *3 * \$B1*. Формулу скопировали в ячейку *E2*. Какой вид приобретет формула в ячейке *E2*?

Решение. В формуле используется смешанная ссылка на ячейку *\$B1*: ссылка на столбец абсолютная, т. е. при копировании столбец не изменится; ссылка на строку относительная, т. е. при копировании номер строки настроится.

В исходной формуле в качестве второго множителя выступают данные ячейки из той же строки, в которой находится и формула (из 1-й). Значит, и в скопированной формуле данные будут браться из одной и той же строки (в данном случае из 2-й).

Ответ: в ячейке *E2* будет записана формула = *3 * \$B2*.

Пример 2. Дан фрагмент электронной таблицы:

	A	B	C	D
1	2	5	3	
2	3	6	4	

В ячейку *D2* введена формула = *A2 * B1 + \$C\$1*. Какое значение появится в ячейке *D2* после вычисления формулы?

Решение. Значение ячейки *A2* — 3, *B1* — 5, *\$C\$1* — 3. Подставив эти значения в формулу, получим: $3 \cdot 5 + 3 = 18$.

Ответ: 18.

Использование функций

Функция — это заранее определенная формула, которая оперирует с одним или несколькими значениями и возвращает одно или несколько значений.

Функция состоит из двух частей

→ **Имя**

Описывает операцию, которую выполняет функция

→ **Аргумент**

Задаёт значение или ячейку, используемую функцией. Заключается в круглые скобки

Например:

$$\text{СУММ}(\text{A1:A6})$$

Между именем функции и ее аргументом не должно быть пробелов.

Некоторые функции не имеют аргументов, но круглые скобки все же необходимо ставить. Например: $= A1 * ПИ()$.

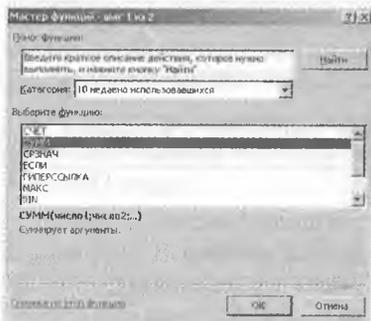
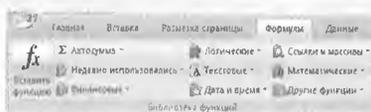
При использовании в функции нескольких аргументов их нужно отделять друг от друга точкой с запятой. Например: функция $= ПРОИЗВ(C1;C2;C5)$ перемножает числа в ячейках $C1$, $C2$ и $C5$.

Аргументы функций могут представлять собой выражения из комбинаций функций.

Например: $= СУММ (SIN (A1 * ПИ ()); 2 * COS (A2 * ПИ ())$. Здесь функции SIN , COS и $ПИ$ сначала вычисляются, а затем используются в качестве аргументов функции $СУММ$.

Вставка функций

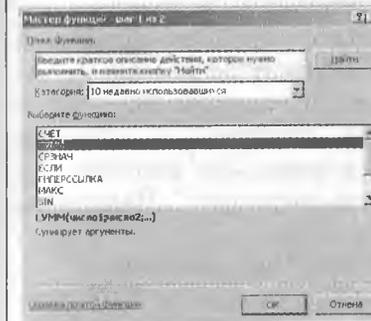
Функцию можно вводить на рабочий лист прямо с клавиатуры.



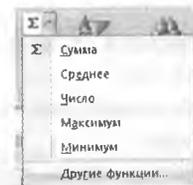
Для ввода функции можно воспользоваться командой **Вставить функцию** из группы **Библиотека функций** вкладки **Формулы**. Excel выведет окно диалога **Мастер функций** — шаг 1 из 2. В этом окне сначала нужно выбрать категорию в одноименном списке, а затем в алфавитном списке **Выберите функцию** указать нужную функцию и нажать кнопку **ОК**. Excel выведет в ячейке и в строке формул знак равенства, имя функции и две круглые скобки, после

чего откроет второе окно диалога мастера функций. В нем находятся поля для ввода аргументов выбранной функции. Если функция имеет переменное число аргументов, то при

вводе дополнительных аргументов это окно диалога увеличивается. Справа от каждого поля аргумента отображается его текущее значение. После ввода всех аргументов нужно нажать кнопку **ОК** — и созданная функция появится в строке формул.



Окно мастера функций можно вызвать также щелчком на кнопке **Вставить функцию** \mathcal{f}_x в строке формул. Кроме того, если ввести знак равенства в ячейку, то поле **Имя** в строке формул заменится раскрывающимся списком, в котором перечислены 10 функций, использовавшихся последними, и пункт **Другие функции**, щелчок на котором откроет окно мастера функций.



Еще один способ вызова окна мастера функций — в группе **Редактирование** вкладки **Главная** раскрыть список опций параметра **Сумма** и выбрать пункт **Другие функции**.

Некоторые встроенные функции

Математические функции

- $= СУММ (числа)$ — функция суммирует множество чисел.
- $= ABS (число)$ — возвращает абсолютное значение числа или формулы.
- $= ОКРУГЛ (число; количество цифр)$ — округляет число, задаваемое аргументом функции, до указанного количества десятичных разрядов.

	<p>= КОРЕНЬ (<i>число</i>) — возвращает положительный квадратный корень из числа. Аргумент должен быть положительным числом.</p> <p>= ПРОИЗВЕД (<i>число1; число2; ...</i>) — перемножает все числа, задаваемые аргументами функции.</p> <p>= LOG (<i>число; основание</i>) — возвращает логарифм положительного числа по заданному основанию. Если аргумент <i>основание</i> не задан, то Excel примет его равным 10.</p> <p>= СТЕПЕНЬ (<i>число; степень</i>) — возводит число в заданную степень Эта функция является эквивалентом оператора $^$. Например, формулы</p> <p>= СТЕПЕНЬ(3;2) и 3^2 равносильны.</p> <p>= SIN (<i>число</i>), = COS (<i>число</i>), = TAN (<i>число</i>) — возвращают синус, косинус, тангенс угла. <i>Число</i> — это угол в радианах</p>
Текстовые функции	<p>= ЗНАЧЕН (<i>текст</i>) — преобразование текста в число. <i>Текст</i> — это числа или даты, введенные в текстовом формате, т. е. заключенные в кавычки.</p> <p>= ДЛСТР (<i>текст</i>) — возвращает количество символов в текстовой строке.</p> <p>= СИМВОЛ (<i>число</i>) — возвращает символ, который соответствует заданному числовому коду ASCII.</p> <p>= КОДСИМВ (<i>текст</i>) — возвращает код ASCII для первого символа аргумента функции</p>
Логические функции	<p>= ЕСЛИ (<i>логическое выражение; значение_если_истина; значение_если_ложь</i>). В качестве аргументов этой функции можно использовать другие функции.</p> <p>= И (<i>логическое значение1; логическое значение2; ...; логическое значение30</i>). Принимает значение ИСТИНА тогда и только</p>

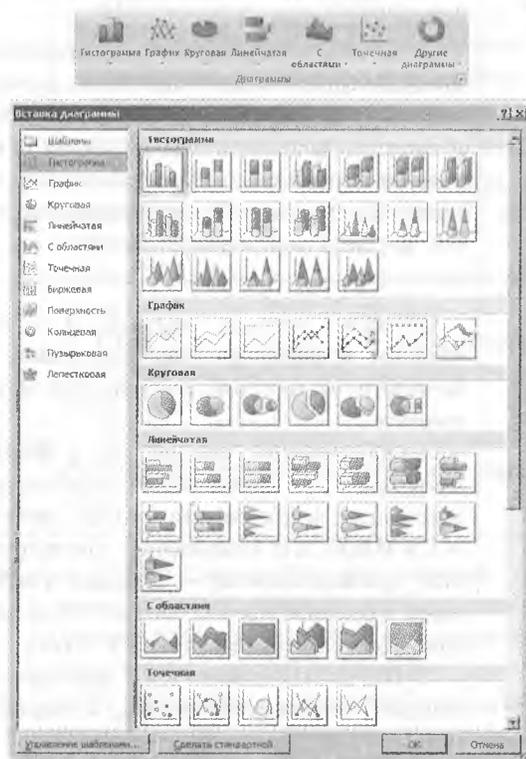
	<p>тогда, когда все аргументы функции имеют значение ИСТИНА.</p> <p>= ИЛИ (<i>логическое значение1; логическое значение2; ...; логическое значение30</i>). Принимает значение ИСТИНА, если хотя бы один аргумент имеет значение ИСТИНА</p> <p>= НЕ (<i>логическое значение</i>). Принимает значение ИСТИНА, если аргумент функции имеет значение ЛОЖЬ, и наоборот</p>
Функции даты и времени	<p>= ДАТА (<i>год; месяц; день</i>) — возвращает десятичное значение даты, которое представляет собой количество дней между базовой и заданной датой.</p> <p>= СЕГОДНЯ () — возвращает десятичное значение текущей даты.</p> <p>= ВРЕМЯ (<i>часы; минуты; секунды</i>) — возвращает десятичное значение, которое представляет собой часть суток между полуночью и заданным временем. Аргумент <i>секунды</i> является необязательным: его можно опустить, но точку с запятой после аргумента <i>минуты</i> все же необходимо поставить</p>
Статистические функции	<p>= СРЗНАЧ (<i>число1; число2;...</i>) — вычисляет среднее арифметическое значение. Функция игнорирует пустые, логические и текстовые ячейки.</p> <p>= МАКС (<i>число1; число2;...</i>), = МИН (<i>число1; число2;...</i>) — возвращают наибольшее и наименьшее значения из набора данных.</p> <p>= СУММЕСЛИ (<i>диапазон; критерий; диапазон суммирования</i>) — каждая ячейка <i>диапазона</i> проверяется на соответствие <i>критерию</i>, прежде чем прибавляется к итогу.</p> <p>= СЧЕТЕСЛИ (<i>диапазон; критерий</i>) — вычисляет количество ячеек, которые удовлетворяют заданному критерию</p>

Визуализация данных с помощью диаграмм и графиков. Построение графиков элементарных функций

Электронные таблицы позволяют визуализировать данные, размещенные на рабочем листе, в виде диаграммы или графика.

Для создания диаграмм и графиков удобно использовать электронные таблицы, оформленные в виде базы данных.

Самый быстрый способ создания диаграммы заключается в следующем: нужно выделить одну ячейку из тех, что содержат исходные данные диаграммы, и нажать клавишу [F11]. Используя установки по умолчанию, Excel создаст диаграмму на отдельном листе.



Диаграммы можно создавать с помощью команд группы *Диаграммы* вкладки *Вставка*. Сначала нужно упорядочить данные, которые будут использоваться для диаграммы, и выделить их. Затем выбрать вид и подвид диаграммы из группы *Диаграммы*. Диаграмма разместится на том же листе как внедренная диаграмма. Далее ее расположение можно изменить.

При создании диаграммы открывается доступ к инструментам для работы с диаграммой: отображается меню *Работа с диаграммами*, содержащее вкладки *Конструктор*, *Макет* и *Формат*. Команды этих вкладок можно использовать для изменения представления данных на диаграммах. Вкладка *Конструктор* используется для отображения рядов данных по строкам или столбцам, внесения изменений в исходные данные, изменения размещения диаграммы, изменения ее типа, сохранения диаграммы в качестве шаблона или выбора предварительно определенных параметров макета и форматирования. Вкладка *Макет* используется для изменения отдельных элементов диаграммы (заголовков, подписей данных и т. д.), использования инструментов рисования, а также для добавления к диаграмме текстовых полей и рисунков. Вкладка *Формат* позволяет добавлять заливку цветом, изменять тип линий или использовать специальные эффекты.

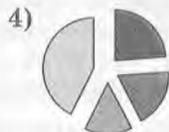
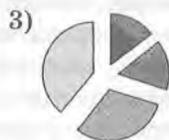
Созданную диаграмму можно рассматривать как формулу. Если внести изменения в данные, использованные при ее создании, то диаграмма изменится. Чтобы отобразить в диаграмме другое множество данных, необходимо выделить область диаграммы и перетащить прямоугольник в рабочем листе, очерчивающий исходные данные, за верхнюю или нижнюю границу на новое место в таблице.

При решении задач нужно помнить, что столбчатые диаграммы обычно отражают абсолютные значения, а круговые — процентный состав. Графики выбирают в тех случаях, когда хотят отобразить изменение данных в зависимости от каких-либо условий.

Пример 1. Дан фрагмент электронной таблицы:

	A	B
1	4	=A1 * A2
2	3	=B1 + 5
4		=B2 - B1
5		=B3 + 2 * A1

После выполнения вычислений была построена диаграмма по значениям диапазона ячеек B1:B4. Какая из диаграмм является верной?

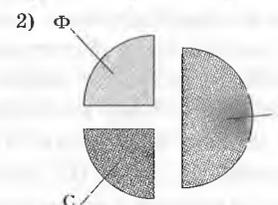
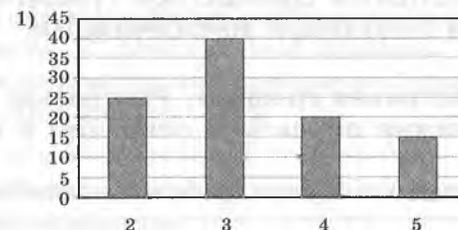


Решение. Сначала нужно вычислить значения в ячейках B1:B4 по указанным формулам:

$$\begin{aligned} B1 &= A1 * A2 = 4 * 3 = 12; \\ B2 &= B1 + 5 = 12 + 5 = 17; \\ B3 &= B2 - B1 = 17 - 12 = 5; \\ B4 &= B3 + 2 * A1 = 5 + 2 * 4 = 13. \end{aligned}$$

Итак, значения ячеек диапазона B1:B4 равны 12; 17; 5; 13. Такому набору значений соответствует диаграмма под номером 1.
Ответ: 1.

Пример 2. В цехе трудятся рабочие трех специальностей: токари (Т), слесари (С) и фрезеровщики (Ф). Каждый рабочий имеет разряд — второй, третий, четвертый или пятый, и только одну специальность. На диаграмме 1 отражено количество рабочих с различными разрядами, а на диаграмме 2 — распределение рабочих по специальностям.



Имеются четыре утверждения:

- А) Все рабочие третьего разряда могут быть фрезеровщиками.
- Б) Все слесари могут быть пятого разряда.
- В) Все токари могут быть четвертого разряда.
- Г) Все рабочие третьего разряда могут быть токарями.

Какое из этих утверждений следует из анализа обеих диаграмм?

- 1) А 2) Б 3) В 4) Г

Решение. По диаграмме 1 найдем общее количество рабочих: $25 + 40 + 20 + 15 = 100$ (человек). Из диаграммы 2 следует, что токари составляют половину всех рабочих, т. е. 50 человек, а слесари и фрезеровщики — по 25 человек.

Проверим утверждение А. Рабочих третьего разряда 40 человек, а фрезеровщиков всего 25. Следовательно, *все* рабочие третьего разряда не могут быть фрезеровщиками. Утверждение ложно.

Проверим утверждение Б. Рабочих пятого разряда всего 15 человек, а слесарей — 25. Поэтому *все* слесари не могут иметь пятый разряд. Утверждение ложно.

Проверим утверждение В. Рабочих четвертого разряда всего 20 человек, а токарей — 50. Поэтому *все* токари не могут иметь четвертый разряд. Утверждение ложно.

Проверим утверждение Г. Токарей всего 50 человек, а рабочих третьего разряда — 40. Значит, все они могут быть токарями. Утверждение истинно.

Ответ: 4.

2.3. ТЕХНОЛОГИЯ ОБРАБОТКИ ГРАФИЧЕСКОЙ И ЗВУКОВОЙ ИНФОРМАЦИИ

Растровая графика. Растровые графические объекты и операции с ними

Общее понятие о растровых графических изображениях

Растровые графические изображения формируются в процессе сканирования рисунков и фотографий, а также при использовании цифровых фото- и видеокамер. Растровое графическое изображение состоит из отдельных точек (*пикселей*), образующих строки и столбцы. Основные свойства пикселей — расположение и цвет. Их значения кодируются двоичным кодом и сохраняются в видеопамяти компьютера.

Качество изображения на экране монитора зависит от пространственного разрешения и глубины цвета.

Пространственное разрешение определяется как произведение количества строк изображения на количество точек в строке.

Глубина цвета измеряется в битах на точку и определяет количество цветов, в которые могут быть окрашены точки изображения. Чем больше пространственное разрешение и глубина цвета, тем выше качество изображения и тем больше объем его файла. В операционных системах предусмотрена возможность выбора необходимого пользователю и технически возможного графического режима.

Растровые изображения чувствительны к **масштабированию**. При уменьшении изображения несколько соседних точек преобразуются в одну, в результате чего теряется четкость мелких деталей изображения. При увеличении изображения увеличивается размер каждой точки и появляется ступенчатый эффект (пикселизация изображения).

Для графических изображений могут использоваться различные **палитры** (наборы) цветов. Количество цветов N в палитре и количество информации I , необходимое для кодирования цвета каждой точки, связаны следующим соотношением:

$$N = 2^I.$$

Например, для черно-белого изображения палитра состоит из двух цветов. С помощью этой формулы можно вычислить, какое количество информации необходимо, чтобы закодировать цвет каждой точки:

$$2 = 2^I \Rightarrow 2^1 = 2^1 \Rightarrow I = 1 \text{ бит.}$$

Информационный объем I_{II} требуемой видеопамяти рассчитывается по формуле

$$I_{II} = I \cdot X \cdot Y,$$

где I — глубина цвета (в битах на точку);

X — количество точек изображения по горизонтали;

Y — количество точек изображения по вертикали.

Цветовые модели

Для кодирования цветов изображения в компьютере используются цветовые модели. **Цветовая модель** (система цветопередачи) — это способ представления различных цветов спектра в виде набора числовых характеристик определенных базовых компонентов.

Цветовая модель RGB

С экрана монитора человеческий глаз воспринимает цвет как сумму излучения трех базовых цветов: **красного (Red)**, **зеленого (Green)** и **синего (Blue)**. Эти цвета называются *основными*. Цвета палитры RGB формируются путем сложения базовых цветов, имеющих различную интенсивность. Цвет (*Color*) в палитре можно определить с помощью формулы

$$Color = R + G + B.$$

Если все базовые цвета имеют максимальную интенсивность, то получается белый цвет, если минимальную — черный. Серый цвет — промежуточный между черным и белым. В нем есть все цветовые составляющие, но они одинаковы и нейтрализуют друг друга. Красный цвет получается при максимальной интенсивности красного и минимальной интенсивности зеленого и синего цветов. Аналогичным образом зеленый

	и синий цвета получаются при максимальной интенсивности соответствующего базового цвета и минимальной интенсивности остальных цветов
Цветовая модель CMYK	<p>Палитра цветов формируется путем наложения базовых цветов: голубого (Cyan), пурпурного (Magenta), желтого (Yellow) и черного (Black). Доля каждого базового компонента задается в процентах (целым числом от 0 до 100). Цвет (<i>Color</i>) в палитре можно определить с помощью формулы</p> $Color = C + M + Y.$ <p>Голубой, пурпурный и желтый цвета называют <i>дополнительными</i>, так как они дополняют основные цвета до белого: голубой дополняет красный цвет, пурпурный — зеленый, а желтый — синий. Данная палитра используется при печати изображений на принтере и основана на восприятии человеком отражаемого света. Изображение на бумаге воспринимается в отраженном свете. Если на бумагу не нанесены краски, то падающий белый свет полностью отражается, и мы видим белый лист бумаги ($White = (C = 0, M = 0, Y = 0)$). Если краски нанесены, то они поглощают определенные цвета.</p> <p>Цвета в палитре формируются путем вычитания из белого цвета определенных цветов. Например, нанесенная на бумагу голубая краска поглощает красный свет и отражает зеленый и синий ($Cyan = W - R = G + B$); желтая краска поглощает синий свет и отражает красный и зеленый ($Yellow = W - B = G + R$).</p>
Цветовая модель HSB	Палитра цветов формируется путем установки значений трех базовых компонентов: оттенка (Hue), насыщенности (Saturation) и яркости (Brightness). Оттенок <i>H</i> определяет цвет в спектре и задается целым числом от 0 до 360 (0 — красный цвет, 360 — фиолетовый).

	Насыщенность <i>S</i> характеризует долю белого цвета, добавленного к выбранному оттенку, и задается в процентах от 0 до 100. При минимальной насыщенности какой-либо оттенок цвета становится серым. Яркость <i>B</i> определяется примесью черного цвета к выбранному оттенку и задается в процентах от 0 до 100. Любой оттенок при минимальной яркости становится черным. Эту модель используют художники при создании компьютерных изображений
--	--

Растровые графические редакторы	
<p>Графические редакторы — специальные программы для создания, редактирования и просмотра графических изображений.</p> <p>С помощью растровых графических редакторов можно обрабатывать цифровые фотографии и отсканированные изображения, повышая при этом их качество путем изменения яркости, контрастности, цветовой палитры, а также удаления дефектов изображения. Кроме того, растровые графические редакторы позволяют создавать новые изображения, применяя к ним различные эффекты преобразования. Одним из самых мощных растровых графических редакторов является Adobe Photoshop.</p>	
Основные возможности растровых графических редакторов	
Создание изображения	Редактирование изображения
<ul style="list-style-type: none"> • Определение области рисования (размер, поля, ориентация холста) • Рисование стандартных графических примитивов (линий, прямоугольников, многоугольников, овалов) 	<ul style="list-style-type: none"> • Выделение области изображения для обработки специальными инструментами • Копирование, перемещение, удаление выделенных областей изображения

<ul style="list-style-type: none"> • Рисование традиционными методами с помощью инструментов рисования (карандаш, кисть, распылитель, заливка, ластик) • Добавление текста и его форматирование • Управление цветом 	<ul style="list-style-type: none"> • Геометрическое преобразование выделенных областей изображения: изменение размера, поворот, наклон, отражение
--	--

Форматы растровых графических файлов

При сохранении графического изображения на внешнем носителе могут использоваться различные способы упорядочения данных в файле, каждый из которых определяет формат (тип) графического файла.

За счет большого количества пикселей в изображении и соответственно большого количества данных, которые нужно сохранять, размеры файлов растровых изображений достаточно велики. Поэтому для растровых графических файлов применяется сжатие. Алгоритм сжатия включается непосредственно в формат графического файла.

Некоторые форматы графических файлов

BMP (Bit MaP image)	Универсальный формат растровых графических файлов. Изображение в этом формате сохраняется попиксельно, без сжатия. Стандартное расширение имени файла — <i>bmp</i> . Поддерживается многими графическими редакторами и рекомендуется для хранения обмена данными с другими приложениями
JPEG (Joint Photographic Expert Group)	Использует эффективные алгоритмы сжатия данных, которые значительно уменьшают размеры файлов. Однако это достигается за счет необратимой потери части данных, а также ухудшения качества изображения.

	Стандартные расширения имени файла — <i>jpg</i> или <i>jpeg</i> . Данный формат целесообразно использовать для хранения многоцветных изображений с плавными переходами между цветами, где потеря качества малозаметна. Формат поддерживается приложениями для различных операционных систем и часто используется для размещения графических изображений на Web-страницах в Интернете
GIF (Graphics Interchange Format)	Самый «плотный» из графических форматов, позволяющих избежать потери информации. Файлы этого формата имеют расширение <i>.gif</i> . В этом формате хранятся и передаются малочетные (до 256 цветов) изображения, например рисованные иллюстрации. Характерная особенность этого формата — возможность создавать необычные эффекты: прозрачность фона и анимацию изображения
TIFF (Tagged Image File Format)	Формат, поддерживаемый всеми основными графическими редакторами; он включает в себя алгоритм сжатия без потерь информации. Файлы этого формата имеют расширение <i>.tif</i> . Они сохраняют изображения с высоким качеством, поэтому широко используются в полиграфии. Формат обеспечивает не очень большую степень сжатия, однако дает возможность сохранять в одном файле дополнительную информацию в невидимых вспомогательных слоях — <i>каналах</i>
PNG (Portable Network Graphic)	Формат, аналогичный формату GIF, однако позволяет использовать в изображении значительно больше цветов. Стандартное расширение имени файла — <i>png</i>

Векторная графика. Векторные графические объекты и операции над ними

Общее понятие о векторной графике

В векторной графике основным элементом изображения является линия.

Линия — это элементарный объект векторной графики. Простейшие объекты объединяются в более сложные, например: объект *четырёхугольник* можно рассматривать как четыре связанные линии, а объект *куб* — либо как 12 связанных линий, либо как 6 связанных четырёхугольников.

Как и все объекты, линии имеют *свойства*: форма линии, ее толщина, цвет, характер (сплошная, пунктирная и т. д.). Замкнутые линии имеют свойство *заполнения*. Внутренняя область замкнутого контура может быть заполнена цветом, текстурой, картой. Незамкнутая линия имеет вершины, которые называются *узлами*.

Хотя объекты векторной графики хранятся в памяти в виде набора параметров, на экран все изображения всегда выводятся в виде точек. Перед выводом на экран каждого объекта программа производит вычисления координат *экранных точек* в изображении объекта. Аналогичные вычисления производятся и при выводе объектов на принтер.

Достоинства векторной графики

1. Небольшой информационный объем файлов. Достаточно сложные композиции изображений, насчитывающие тысячи объектов, расходуют лишь десятки и сотни килобайт.

2. Легко решаются вопросы *масштабирования*. Если линии задана толщина, равная 0,15 мм, то, сколько бы мы ни увеличивали или ни уменьшали рисунок, эта линия все равно будет иметь только такую толщину, поскольку это одно из жестко закрепленных свойств объекта. Это свойство векторной графики широко используется в картографии, в конструкторских системах автоматизированного проектирования (САПР), в автоматизированных системах архитектурного проектирования.

Векторные графические редакторы

Векторные графические редакторы — программы, предназначенные для работы с векторными изображениями. Их применяют в тех случаях, когда основным требованием к изображению является высокая точность формы. Такая задача возникает при разработке логотипов компаний, при художественном оформлении текста (например, журнальных заголовков или рекламных объявлений), а также во всех случаях, когда иллюстрация является чертежом, схемой или диаграммой, а не рисунком. Векторная графика также лежит в основе flash-анимации. Примеры редакторов: Adobe Illustrator, Macromedia Freehand, CorelDraw.

Основные возможности векторных графических редакторов

Создание изображения	Редактирование изображения
<ul style="list-style-type: none"> • Определение области рисования (размер, поля, ориентация холста) • Рисование стандартных графических примитивов (линий, прямоугольников, многоугольников, овалов) • Добавление текстовых областей и выносок • Изменение видимости объектов путем изменения порядка размещения их слоев (каждый графический объект изображается в своем слое, а весь рисунок состоит из множества слоев) • Градиентная заливка объектов • Задание степени прозрачности объекта 	<ul style="list-style-type: none"> • Выделение отдельного объекта или группы объектов в рисунке • Группировка нескольких объектов в один новый объект • Выравнивание объектов с помощью сетки, которую можно настраивать • Копирование, перемещение, удаление выделенных объектов • Геометрическое преобразование выделенных объектов: изменение размера, поворот, наклон, отражение

Форматы векторных графических файлов	
Разнообразие форматов векторной графики значительно меньше, чем растровой графики, и практически каждый векторный редактор использует свой собственный формат сохранения данных.	
Некоторые форматы векторных графических файлов	
WMF (Windows MetaFile)	Универсальный формат векторных графических файлов для приложений Windows. Используется для хранения коллекции графических изображений Microsoft Clip Gallery. Возможные расширения файлов WMF, EMF, WMZ, EMZ
CGM (Computer Graphic Metafile)	Широко используется как стандартный формат векторных графических данных в Интернете
EPS (Encapsulated PostScript)	Формат, поддерживаемый программами для различных операционных систем. Рекомендуется для создания иллюстраций в настольных издательских системах
CDR (CorelDraw files)	Оригинальный формат файлов векторного графического редактора CorelDraw. Изображение в файле может состоять из нескольких страниц. Формат позволяет сохранять не только векторную графику, но также текст и растровые изображения. Максимальный размер рисунка 45×45 м. Возможны расширения файлов CDR или CDT
AI (Adobe Illustrator files)	Оригинальный формат файлов векторного графического редактора Adobe Illustrator. Сохраняет в файле только одну страницу, максимальный размер рисунка 3×3 м

SVG
(Scalable
Vector
Graphics)

Универсальный формат двухмерной графики; позволяет сохранять в файле текст, графические изображения и анимацию. Файлы могут дополнительно сжиматься программами-архиваторами. Формат обрабатывается практически всеми векторными графическими редакторами. Широкое применение получил в инженерной графике и при разработке Web-сайтов

2.4. СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ЦИФРОВЫХ ЗВУКОЗАПИСЕЙ

Звук. Запись и воспроизведение звука

Звук — это продольная волна сжатия-расширения, распространяющаяся в воздушной, водной или какой-либо другой среде (механические колебания). Звук характеризуется **интенсивностью**, которую человек воспринимает как громкость, и **частотой**, воспринимаемой как тон. Чем больше интенсивность, тем громче звук; чем выше тон, тем выше частота

Для преобразования звуковых колебаний в электрические применяют **микрофоны**. Чтобы микрофон можно было подключить к компьютеру, компьютер должен быть оснащен **звуковой картой**. В последнее время звуковые карты интегрированы в материнские платы, однако лучшее качество обеспечивают дискретные и внешние карты

Звуковая волна по своей природе непрерывна. Для ее обработки в компьютере производится дискретизация по времени и по уровню. В результате этого процесса непрерывная волна заменяется совокупностью цифровых отсчетов громкости сигнала в дискретные равноотстоящие моменты времени. Эту задачу выполняет **АЦП** (аналогово-цифровой преобразователь) звуковой карты. Чем больше разрядность АЦП, тем точнее передается форма звуковой волны, тем меньше искажения и тем больше динамический диапазон

Прямая запись звуковой волны приводит к большим объемам файлов. Поэтому для аудиофайлов было изобретено множество форматов

Форматы звуковых файлов

Форматы без потери качества (*lossless*): WAV, APE, FLAC

Форматы с потерей качества (*lossy*): MP3, OGG, WMA и др.

Звуковые редакторы

Звуковые редакторы — специальные программы для создания и редактирования цифровых звукозаписей

Примеры звуковых редакторов: **Adobe Audition, Sound Forge**

Некоторые возможности звуковых редакторов:

- запись звука;
- воспроизведение звука;
- синтезирование звука;
- редактирование звука (копирование, перемещение и удаление части звуковой дорожки, реставрация звука, нелинейный монтаж, создание всевозможных акустических эффектов).

Запись звука

Современные звуковые редакторы позволяют записывать аудиофайлы в различных форматах в зависимости от того, как они далее будут использованы (на чем будут воспроизведены).

Чтобы звук попал в компьютер, необходимо подключить источник сигнала к звуковой карте, например: микрофон — к микрофонному входу, другой источник — к линейному входу звуковой карты. Далее необходимо настроить звуковую карту, т. е. выбрать источник и уровень сигнала. Для этого используется стандартная программа **Регулятор громкости**, которая входит в состав операционной системы Windows.



При выборе команды *Параметры/Свойства* откроется окно *Свойства*, в котором нужно включить переключатель *Запись*, указать нужный вход (Microphone или Line-in) и установить необходимый уровень записи. Далее можно использовать любой звуковой редактор. При использовании звуковых редакторов новый файл создается традиционной командой *File/New*. При этом потребуется установить частоту дискретизации и глубину кодирования звукового сигнала, а также режим записи (стерео или моно). Для музыкального сигнала лучше использовать параметры компакт-диска (44100 Гц, 16 бит, два канала); если требования к качеству звука выше, можно повысить частоту дискретизации до 48 000 Гц. Речь с микрофона в нестудийных условиях с приемлемым качеством можно записывать и с частотой дискретизации 8000 Гц, 8 бит, один канал. Дальнейшие действия выполняются в соответствии с инструкцией к конкретному редактору.

Воспроизведение звука

Для того чтобы прослушать звуковой файл, нужно подключить к линейному выходу звуковой карты колонки или наушники. Сравнительно недавно появились компьютерные колонки со встроенным цифро-аналоговым преобразователем (ЦАП), питание и сигнал в которые подается через шину USB. В этом случае можно воспроизводить аудиофайлы на компьютере без звуковой карты. Компьютерный звук воспроизводится специальной программой-плеером, которая управляет ЦАП звуковой карты. В состав операционной системы Windows входит стандартный проигрыватель **Windows Media Player**; из аналогичных продуктов других фирм очень популярна программа **WinAmp**. Эти программы позволяют воспроизводить аудиофайлы практически любого формата.

2.5. КОМПЬЮТЕРНЫЕ ПРЕЗЕНТАЦИИ: ТИПЫ СЛАЙДОВ, МУЛЬТИМЕДИА ЭФФЕКТЫ, ОРГАНИЗАЦИЯ ПЕРЕХОДОВ МЕЖДУ СЛАЙДАМИ

Понятие компьютерной презентации

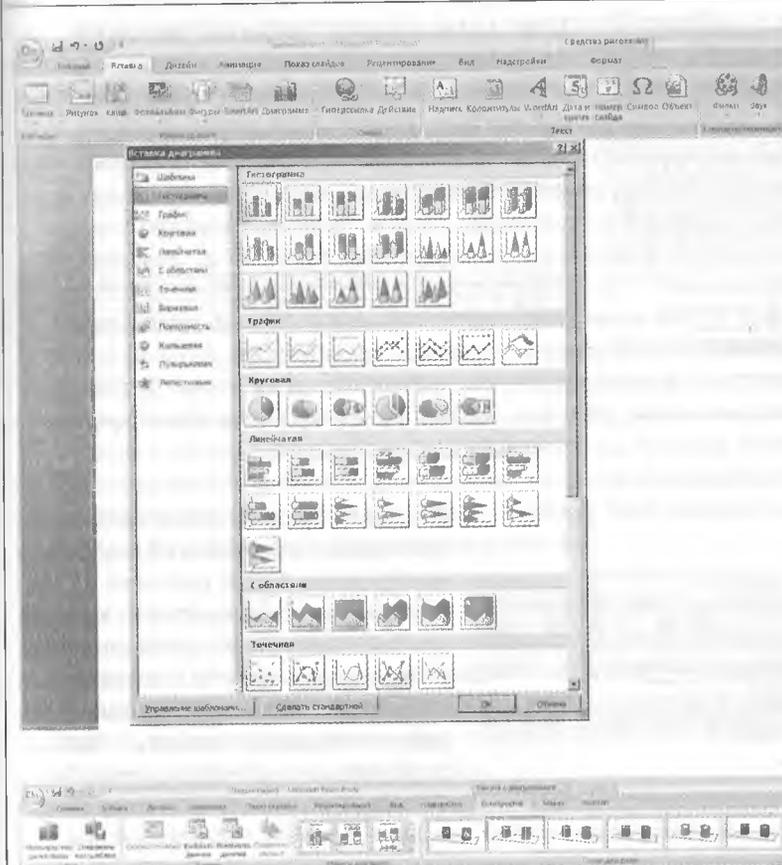
Компьютерная презентация — ряд последовательно демонстрируемых в процессе урока, лекции, конференции, совещания и т. д. изображений — слайдов, содержащих текстовую и графическую информацию.

Основные типы слайдов

Текстовые слайды

К текстовым слайдам относятся заголовки, маркированные списки и текст в две колонки. В слайдах других видов текст комбинируется с такими элементами, как картинки, диаграммы или видеоклипы. На ленте появляется дополнительная панель *Средства рисования* с вкладкой *Формат*, команды которой позволяют отформатировать текст: задать стиль, цвет контура и заливки, добавить эффекты (тень, отражение, рельеф и др.), выровнять или развернуть текст, изменить его размер и переместить на задний или передний план.

Слайды с диаграммами



Для того чтобы поместить на слайд диаграмму, необходимо воспользоваться командой *Диаграмма* из вкладки *Вставка*. При помещении диаграммы необходимо помнить, что **столбчатые** диаграммы обычно отражают абсолютные значения, а **круговые** — процентный состав. **Графики** выбирают в тех случаях, когда хотят отобразить изменение данных в зависимости от каких-либо условий.

Когда диаграмма добавлена на слайд, на ленте появляется дополнительная панель *Работа с диаграммами*, содержащая вкладки *Конструктор*, *Макет* и *Формат*.

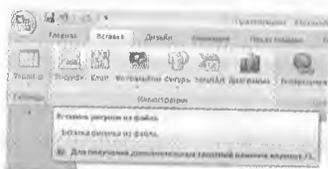
Слайды с таблицами



При активизации таблицы на ленте появляется дополнительная панель *Работа с таблицами*, содержащая вкладки *Конструктор* и *Макет*, команды которых позволяют настроить внешний вид таблицы и ее структуру.



Слайды с рисунками



На слайд можно поместить предлагаемые программой образцы изображений или рисунок из любого файла пользователя, причем изображение можно как внедрить в документ, так и связать его с файлом. На ленте появляется дополнительная панель *Работа с рисунками*, содержащая вкладку *Формат*.



Порядок создания презентации в программе PowerPoint из пакета Microsoft Office 2007

При создании презентации в программе PowerPoint необходимо придерживаться следующей последовательности.

1. Выбрать *тему* слайда на вкладке *Дизайн*. От выбранной темы зависит вид презентации: цветовая схема слайда, его формат, графические элементы. Щелчком правой кнопки по выбранному образцу нужно вызвать контекстное меню и выбрать, к каким слайдам его применить.
2. Определить нужные элементы и *разместить* их на слайде. Для этого можно воспользоваться предлагаемыми программой макетами, например, маркированным списком, диаграммой или таблицей. Чтобы выбрать макет слайда, нужно воспользоваться командой *Макет* из группы *Слайды* вкладки *Главная*, а затем щелкнуть на нужном макете, чтобы применить его к текущему слайду. Выбранный макет необходимо заполнить содержимым.
3. Окончательно оформить слайд. Для добавления анимации на слайд нужно выбрать желаемый стандартный эффект из списка *Анимация* или выполнить команду *Настройка анимации* и задать дополнительные параметры. Для изменения цветовой схемы слайда следует повторно выбрать образец темы на вкладке *Дизайн*.
4. Аналогичным образом создать остальные слайды, воспользовавшись командой *Создать слайд* из группы *Слайды* вкладки *Главная*, и оформить их.
5. Проверить порядок слайдов. Общую картину презентации отображает режим *Сортировщик слайдов*. В этом режиме можно изменять порядок слайдов перетаскиванием мыши, удалять их или копировать, а также переопределять размер слайдов в раскрывающемся меню *Масштаб*. Переход в данный режим осуществляется с помощью команды *Сортировщик слайдов* из группы *Режимы просмотра презентации* вкладки *Вид*. В этом режиме можно добавить переходы между слайдами и анимацию.

6. Задать переходы между слайдами. Простые переходы между слайдами помогают при смене кадров удерживать внимание аудитории. Несложная анимация, например, постепенное затухание звука или изображения, может ненавязчиво, не бросаясь в глаза, представлять элементы слайда по мере обращения к ним. Чтобы добавить переход, нужно выбрать желаемый эффект (жалюзи, наплыв, прорезание и т. д.) в группе *Переход к этому слайду* вкладки *Анимация*, установить скорость перехода; добавить звук.

7. Сохранить презентацию командой *Сохранить* или *Сохранить как*. Для открытия и закрытия презентации служат соответственно команды *Открыть* и *Закрыть*. Распечатать презентацию можно с помощью команды *Печать*. Все эти команды находятся в окне, открываемом щелчком по кнопке *Office*.

8. Просмотреть созданную презентацию. Для просмотра презентации с первого слайда нужно выполнить команду *С начала* из группы *Начать показ слайдов* вкладки *Показ слайдов* или нажать клавишу [F5]. Чтобы увидеть следующий слайд, необходимо нажать клавишу пробела, клавишу [PgDn], [Enter] или просто щелкнуть мышью. Чтобы просмотреть слайд-шоу начиная с текущего слайда, следует выбрать команду *С текущего слайда* из той же группы. Отменить показ слайдов можно клавишей [Esc]. Режим показа презентации и режим редактирования существуют независимо друг от друга. Во время демонстрации слайдов все меню редактирования недоступны и на экране не отображаются.

Автоматический показ слайдов

PowerPoint позволяет задать автоматический показ слайдов (без щелчков мышью для перехода к следующему слайду). Можно предварительно отрепетировать показ презентации. По команде *Настройка времени* из группы *Настройка* вкладки *Показ слайдов* на экране появится первый слайд и откроется панель инструментов *Репетиция*. Теперь нужно отрепетировать

свою речь при отображении каждого слайда. Когда вы будете готовы перейти к следующему слайду, нажмите любую из клавиш, выполняющих эту функцию, или щелкните по кнопке *Далее* панели инструментов *Репетиция*. Те же действия следует выполнять на протяжении всего показа слайдов. После его завершения программа отобразит общее время показа и задаст вопрос, нужно ли сохранить время смены для каждого слайда и использовать его при показе презентации. Щелчок по кнопке *Да* запишет время и создаст автоматический показ слайдов.

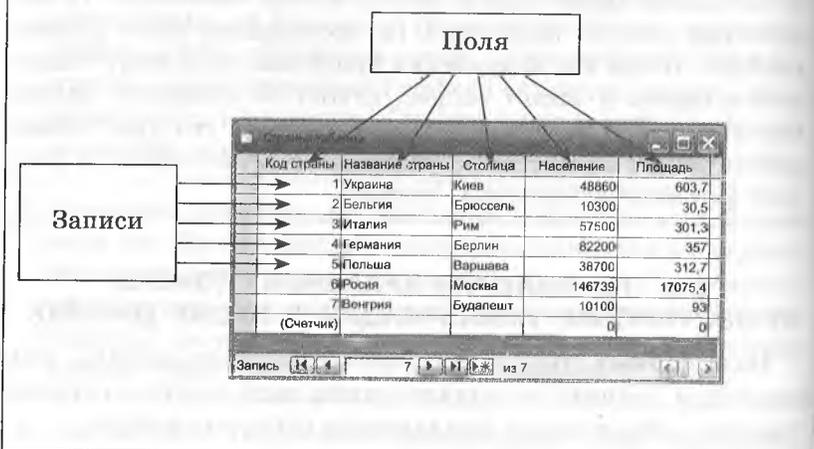
2.6. ТЕХНОЛОГИЯ ХРАНЕНИЯ, ПОИСКА И СОРТИРОВКИ ИНФОРМАЦИИ В БАЗАХ ДАННЫХ

База данных (БД) — это информационная модель, позволяющая хранить в упорядоченном виде данные о группе объектов, обладающих одинаковым набором свойств.

Структура базы данных (поля и записи)

Основные типы организации данных в БД	
Иерархическая БД	Элементы в записи упорядочены. Один из элементов считается главным, остальные — подчиненными
Сетевая БД	Более гибкая структура, т. к. в ней дополнительно к вертикальным связям устанавливаются горизонтальные связи
Табличная БД	Информация организована в виде таблиц. Каждая строка таблицы содержит информацию об одном отдельном объекте описываемой в БД предметной области, а каждый столбец — определенные характеристики (свойства, атрибуты) этих объектов. В БД столбцы называются <i>полями</i> , а строки — <i>записями</i> . Поля образуют <i>структуру</i> БД, а записи составляют информацию, которая в ней содержится

Поля — основные элементы структуры БД



Основные свойства поля

Размер поля	Выражается в знаках (или символах). От длины поля зависит, сколько информации в нем может поместиться. Поскольку символы кодируются одним или двумя байтами, можно считать, что длина поля измеряется в байтах
Имя поля	Имена полей должны быть уникальными, иначе компьютер запутается в содержимом полей
Подпись	Это та информация, которая отображается в заголовке столбца. Если подпись не задана, то в заголовке отображается имя поля. Разные поля могут иметь одинаковые подписи
Формат поля	Устанавливает формат данных

Основные типы полей
(тип поля определяется типом данных, которые оно содержит)

Текстовое	Для ввода текста. Максимальный размер — 255 символов
Числовое	Для ввода числовых данных
Дата/время	Для ввода даты и времени в определенном формате
Логическое	Для ввода логических данных, имеющих только два значения (да — 1, нет — 0). Его длина всегда равна 1 байту
Денежное	Для ввода чисел в денежном формате
Поле объекта OLE	Для хранения картинок, музыкальных клипов, видеозаписей
Счетчик	Это числовое поле, имеющее свойство автоматического наращивания. При вводе новой записи в поле автоматически вводится число, на единицу большее, чем значение этого поля в предыдущей записи
Поле MEMO	Используется, если нужно вставить длинный текст. В него можно поместить до 65535 символов (64 Кбайт). Особенность этого поля заключается в том, что в нем хранятся не сами данные, а только указатель на то место, где они размещены
Гиперссылка	Ссылка на информационный ресурс в Интернете

Пример. В таблице представлен фрагмент базы данных о клиентах некоторой фирмы.

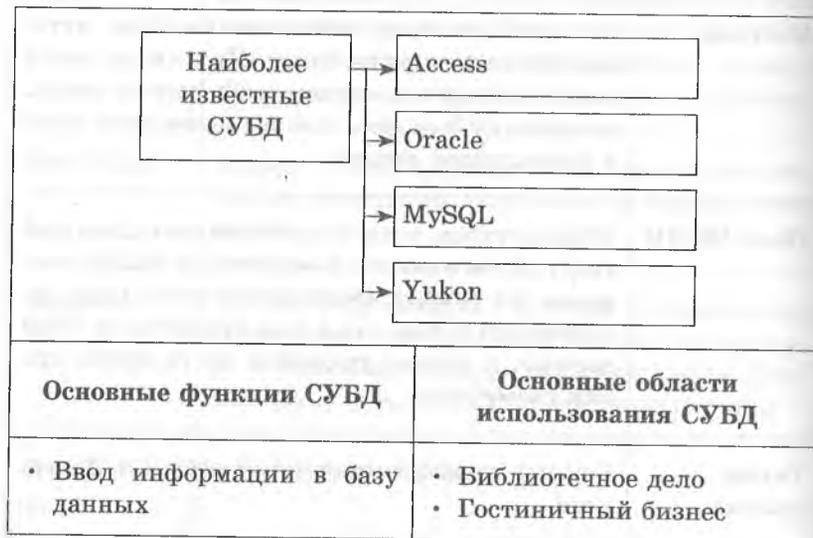
Код_Клиента	Фамилия	Адрес	Телефон	E-mail
K1216P	Карпов	ул. Кирова, 25	2892316	KDA@mail.ru
M1347П	Маслов	ул. Королева, 12	7937847	MaslovN63@rambler.ru

Сколько полей и сколько записей в данной базе?

Решение. Поля — это столбцы базы данных, а записи — это строки. Следовательно, в данной базе 5 полей (Код_Клиента, Фамилия, Адрес, Телефон, E-mail) и 2 записи (о клиентах Карпове и Маслове).

Системы управления базами данных (СУБД)

Система управления базами данных (СУБД) — это приложение для создания баз данных с целью хранения и поиска необходимой информации.



- Возможность редактирования информации
- Удаление ненужной информации
- Защита данных от разрушения
- Поиск информации по заданным критериям
- Автоматическое упорядочение информации в соответствии с определенными требованиями
- Обеспечение доступа к данным нескольких пользователей одновременно
- Защита от несанкционированного доступа к данным
- Бухгалтерский учет
- Маркетинговые исследования
- Документооборот
- Управление предприятием
- Поддержка web-сайтов
- Электронные справочники

Работа в программной среде СУБД предполагает реализацию двух этапов

Создание базы данных

Управление базой данных

Инструменты системы управления базой данных

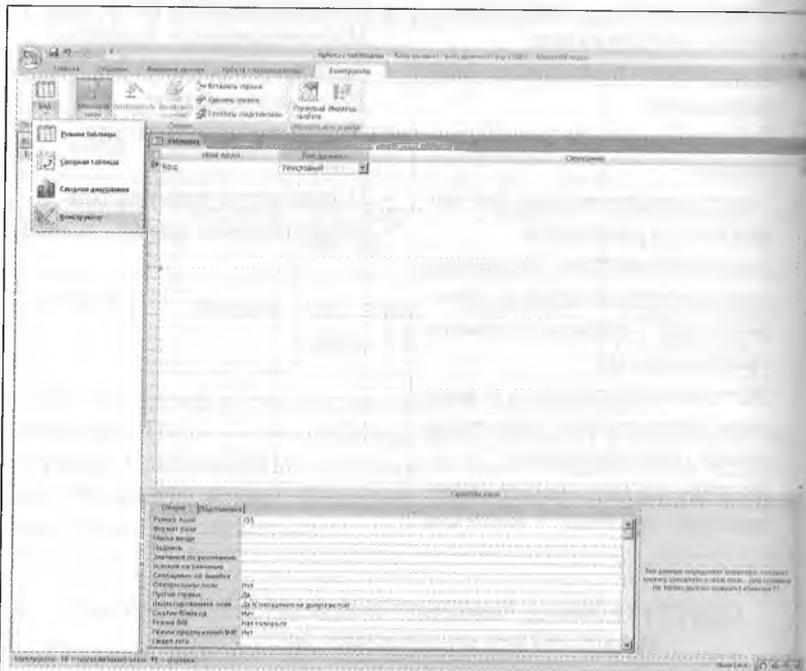
→ Инструменты для создания таблиц

→ Инструменты для управления видом представления данных

→ Инструменты для обработки данных

→ Инструменты для вывода данных

Инструменты СУБД Access для создания таблиц



Работа с базой данных начинается с создания таблицы. Для создания структуры таблицы используется режим конструктора. В этом режиме открывается *бланк структуры таблицы*, в котором необходимо:

- 1) ввести имена полей в графе *Имя поля*;
- 2) выбрать из списка типы данных полей в графе *Тип данных*;
- 3) задать ключевое поле;
- 4) описать дополнительные свойства полей на вкладке *Общие*.

Для наполнения таблицы данными нужно перейти в режим таблицы.

Инструменты СУБД Access для управления видом представленных данных

Способы представления информации в базах данных

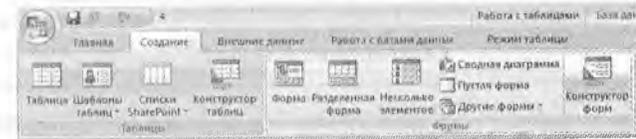
Таблица

Форма

Таблица — наиболее распространенный способ представления данных. Однако он не самый удобный.

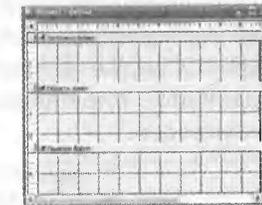
Наиболее удобным способом просмотра и ввода данных является **форма**.

Форма может быть создана с помощью **конструктора форм** или с помощью **мастера**. Наиболее быстрый и удобный вариант — с помощью **мастера**. Конструктор форм позволяет изменить дизайн формы.



Составные части формы

- Заголовок формы
- Область данных
- Примечания



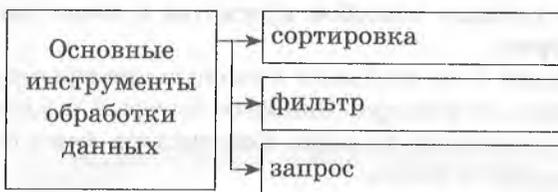
Заголовок формы и **примечания** предназначены для постоянного отображения информации.

В **области данных** располагаются графические элементы для введения и отражения данных таблиц запросов. Эти графические элементы называются **элементами управления**.

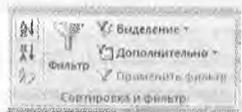
Основные элементы управления

	Надпись		Кнопка
	Текстовое поле		Список
	Переключатель		Присоединенная рамка объекта
	Флажок		

Инструменты СУБД Access для обработки данных



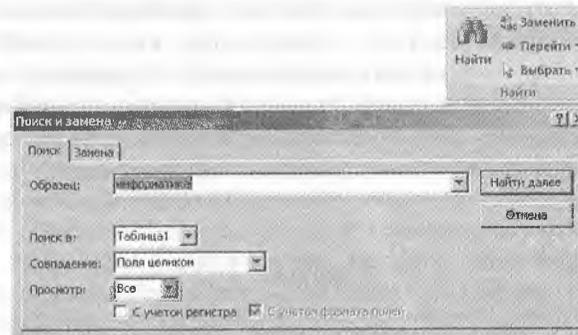
Сортировка данных



Сортировка — это упорядочение данных по некоторому признаку. Различают сортировку по возрастанию и по убыванию. Для числовых значений сортировка означает ранжирование по значению, а для текстовых — упорядочение по алфавиту.

В СУБД Access сортировка данных осуществляется только по одному полю. Каждая новая сортировка отменяет результаты предыдущей. Вложенные сортировки выполняются с помощью запросов.

СУБД Access позволяет производить поиск записей, в которых значения определенного поля полностью или частично совпадают с заданной величиной. Поиск информации в таблице осуществляется следующим образом. Вначале нужно установить курсор в поле, в котором будет выполняться поиск, после чего открыть на ленте вкладку *Главная* и в группе *Найти* выбрать команду *Найти* . Откроется окно *Поиск и замена*.



В поле *Образец* необходимо ввести значение, которое следует найти. Для этого можно использовать подстановочные знаки (маску):

«*» — любой символ в начале или в конце;

«?» — любой символ в любом месте;

«[]» — любой символ из заключенных в скобки (например, маска *b[ae]ll* найдет *ball*, *bell* диапазона (например, *b[a-c]d* найдет *bad*, *bbd*, *bcd*);

«#» — любая цифра.

После задания образца для поиска необходимо задать остальные параметры поиска и щелкнуть на кнопке *Найти далее*. Если заданное значение будет найдено, оно выделится цветом. Чтобы продолжить поиск, необходимо снова щелкнуть на кнопке *Найти далее*. Если данных больше не будет найдено, программа выдаст соответствующее сообщение. Вкладка *Замена* этого окна позволяет не только найти какую-либо информацию, но и заменить ее на другую.

Фильтр

Выбор из базы тех записей, которые удовлетворяют требованиям пользователя, осуществляется также с помощью **фильтров**. Для задания фильтра нужно открыть на ленте вкладку *Главная* и в группе *Сортировка и фильтр* воспользоваться командой *Фильтр*. Команды этой группы *Выделение* и *Дополнительно* предлагают несколько дополнительных

видов фильтров: например, фильтр «по выделенному» позволяет отобразить записи, содержащие выделенный фрагмент; фильтр «исключить выделенное» — скрывает записи, содержащие выделенный фрагмент; расширенный фильтр позволяет задать более сложные условия отбора записей, а также сортировку по нескольким параметрам.

Запрос

Запрос — это мощное средство обработки данных, соединяющее в себе возможности, предоставляемые сортировкой и фильтрацией.



Для создания запроса на ленте нужно открыть вкладку *Создание* и в группе *Другие* выбрать команду *Конструктор запросов*. При этом на ленте появится вспомогательная вкладка *Работа с запросами/Конструктор*. Программа также выведет бланк запроса и окно *Добавление таблицы*. Сначала следует задать таблицы или запросы, на которых будет основываться новый запрос.

Это выполняется указанием имени таблицы (запроса) и щелчком по кнопке *Добавить* в окне *Добавление таблицы*. Далее нужно заполнить строки нижней панели бланка, которые определяют структуру запроса, т. е. результирующей таблицы.

Строку *Поле* заполняют перетаскиванием названий полей из таблиц в верхней части бланка. Каждому полю будущей результирующей таблицы соответствует один столбец бланка запроса по образцу. Строка *Имя таблицы* заполняется автоматически при перетаскивании поля. Щелчок по кнопке *Сортировка* выведет кнопку раскрывающегося списка, содержащего виды сортировки.

Флажок *Вывод на экран* указывает, нужно ли выводить поле на экран.

В строке *Условие отбора* записывают тот критерий, по которому выбирают записи для включения в результирующую таблицу. По каждому полю можно создать свое условие отбора. Для формирования условий можно использовать логические операции (*И*, *ИЛИ* и др.)

Запуск запроса осуществляется выбором команды *Выполнить* из группы *Результаты* на вкладке *Работа с запросами/Конструктор*.

Пользователь, не имевший отношения к созданию базы, не может изменить параметры, по которым делается запрос на выборку. Но часто пользователю необходимо предоставить возможность выбора того, что он хочет найти в таблицах. Для этого существует специальный запрос — **запрос с параметром**. В строке *Условие отбора* бланка запроса в нужном поле следует ввести команду языка SQL:

LIKE [текст]

После запуска запроса на экран выведется окно, содержащее *текст*. В этом окне пользователю предлагается ввести параметр. По щелчку на кнопке *ОК* появится результирующая таблица

С помощью запросов можно производить **вычисления**. Поле, содержимое которого является результатом расчета по содержимому других полей, называется *вычисляемым* полем. Оно может существовать только в результирующей таблице. В исходных (базовых) таблицах такое поле не создается, и таблицы не меняются.

Для создания запроса, производящего вычисления, служит тот же самый бланк запроса. Разница лишь в том, что в одном из столбцов записывают формулу. В нее входят заключенные в квадратные скобки названия полей, участвующих в расчете, а также знаки математических операций. Если формула большая, то с помощью клавиш *[Shift]-[F2]* можно открыть вспомогательное окно *Область ввода*. В него можно ввести сколь угодно длинную формулу, а затем щелчком на кнопке *ОК* перенести ее в бланк запроса по образцу.

Пример 1. К нескольким файлам применялись различные архиваторы и в таблицу записывались исходный размер файлов и размеры архивов. Ниже приведен фрагмент этой таблицы:

Имя файла	Размер	ZIP	RAR	ARJ
Text1.doc	285	114	106	112
Ref.doc	843	42	34	41

Требуется отобразить файлы, исходный размер которых составлял более 2 Мбайт, а при использовании WinRAR уменьшился более чем в 4 раза. Какое условие нужно использовать для формирования запроса?

- 1) (Размер > 2000) И (Размер/RAR > 4)
- 2) (Размер > 2048) И (RAR < 256)
- 3) (Размер > 2048) ИЛИ (Размер/RAR > 4)
- 4) (Размер > 2048) И (Размер/RAR > 4)

Решение. Исходя из условия задачи, в условии запроса должна использоваться логическая операция И (логическое умножение), поэтому вариант 3) не подходит. Кроме того, следует учесть, что 1 Мбайт = 1024 байта. Запрос с условием 1) может отобразить записи о файлах с исходным размером менее 2 Мбайт, следовательно, также не подходит. Правильный вариант — 4).

Ответ: 4).

Пример 2. В таблице представлен фрагмент базы данных, содержащей информацию о странах, их площади в тыс. кв. км (поле *Площадь*), численности населения в тыс. человек (поле *Население*), расположении в части света (поле *ЧС*):

№	Страна	Столица	Площадь	Население	ЧС
1	Бельгия	Брюссель	30.5	10289	Европа
2	Коста-Рика	Сан-Хосе	51.1	3896	Северная Америка
3	Израиль	Тель-Авив	20.8	6116	Азия
4	Дания	Копенгаген	43.1	5384	Европа

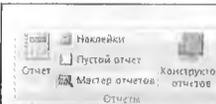
Какое условие нужно использовать для формирования запроса, отбирающего все европейские страны площадью менее 30 тыс. кв. км с численностью населения более 10000 тыс. человек?

- 1) (Площадь < 30) И (Население > 10000)
- 2) (ЧС = Европа) И (Площадь < 30)
ИЛИ (Население > 10000)
- 3) (Площадь < 30) И (Население > 10000) ИЛИ (ЧС = Европа)
- 4) (ЧС = Европа) И (Площадь < 30) И (Население > 10000)

Решение. Исходя из условия задачи, в условии запроса должна использоваться логическая операция И (логическое умножение), поэтому варианты 2) и 3) не подходят. Условие 1) также неверно, т. к. в нем не задано расположение страны (Европа), т. е. по такому запросу будут отобраны страны, расположенные в Африке, Азии и т. д. Правильный вариант — 4).

Ответ: 4).

Инструменты СУБД Access для вывода данных



Наиболее удобной формой представления информации для вывода на печать является отчет.

Удобнее всего создавать отчет с помощью мастера, а корректировать — в режиме конструктора

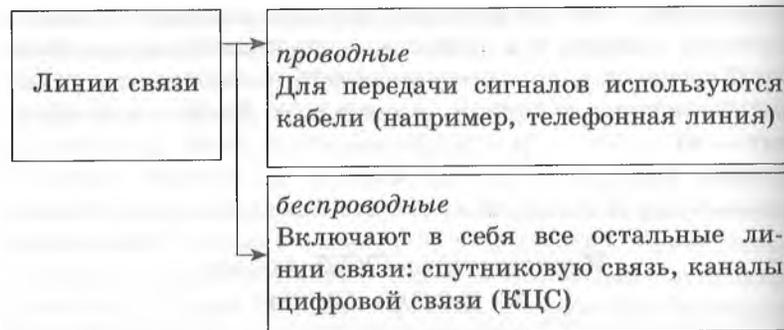
2.7. СРЕДСТВА И ТЕХНОЛОГИИ ОБМЕНА ИНФОРМАЦИЕЙ С ПОМОЩЬЮ КОМПЬЮТЕРНЫХ СЕТЕЙ

Базовые принципы организации и функционирования компьютерных сетей

Компьютерная сеть (КС) — это совокупность компьютеров, соединенных линиями связи и оснащенных коммуникационным оборудованием и программным обеспечением.

Коммуникационное, или сетевое оборудование — это периферийные устройства, которые осуществляют преобразование сигналов, используемых в компьютере, в сигналы, передаваемые по линиям связи, и наоборот. При использовании телефонных линий связи применяют *модемы*, подключаемые к компьютеру через стандартный последовательный порт. При использовании других линий связи применяют *сетевые адаптеры*.

Линия связи — это оборудование, с помощью которого осуществляется объединение компьютеров в сеть.

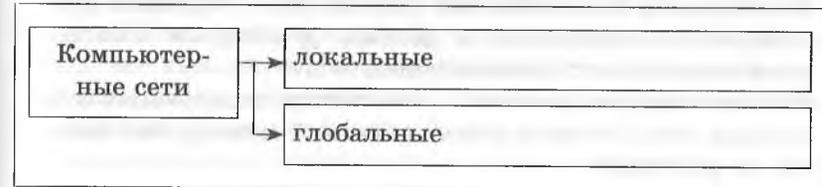


Коммуникационное, или сетевое программное обеспечение — это набор программ, обеспечивающий работу сетевого оборудования и обмен информацией между компьютерами в сети.

Сетевое программное обеспечение

«низкого» уровня	«высокого» уровня
Управляет сетевым оборудованием с целью преобразования сигналов из одного вида в другой. Эти программы ничего «не знают» о структуре передаваемой информации	Распознает и обрабатывает информацию в зависимости от ее характера и способа организации

Локальные и глобальные вычислительные сети

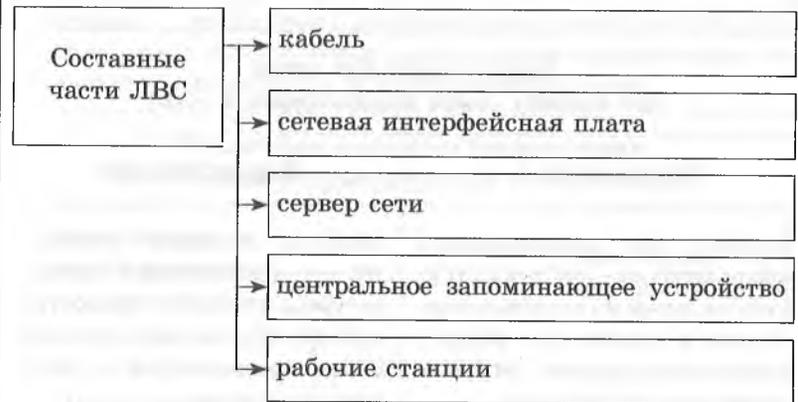


Локальные вычислительные сети (ЛВС)

Локальная вычислительная сеть (ЛВС) объединяет компьютеры, расположенные на небольшом расстоянии друг от друга, и представляет собой замкнутую систему.

Локальные вычислительные сети позволяют:

- совместно использовать аппаратные ресурсы;
- совместно использовать программные ресурсы;
- создавать и совместно использовать информационные ресурсы для работы пользователей над общими задачами;
- централизовать усилия по информационной безопасности.



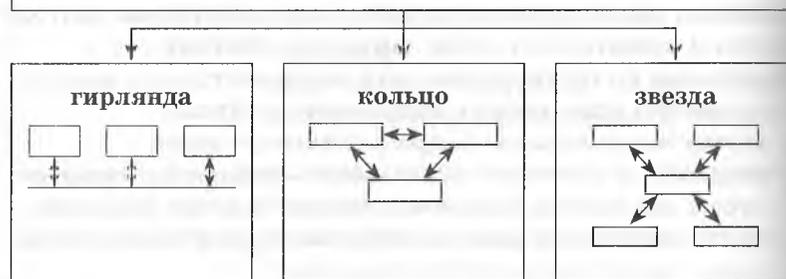
К кабелю передачи данных подключено каждое устройство в сети, поэтому возможен обмен информацией между ними. **Сетевая интерфейсная плата** — специальное аппаратное средство для эффективного взаимодействия ПК в сети. **Сервер сети** — компьютер, предоставляющий свои ресурсы для совместного использования.

Центральное запоминающее устройство — жесткий диск, содержащий программы и данные, к которым возможен совместный доступ пользователей сети.

Рабочая станция (клиент) — компьютер, пользующийся ресурсами сети. Ресурсы рабочих станций серверу, как правило, не доступны.

Топология ЛС — это способ объединения компьютеров между собой в сеть.

Основные топологии локальных вычислительных сетей



Виды локальных сетей (по способу связи компьютеров в сети)

Одноранговые	Иерархические
Состоят из равноправных компьютеров — рабочих станций, каждый из которых имеет имя и пароль для входа в компьютер в момент загрузки операционной системы	Имеется мощный компьютер — <i>выделенный сервер</i> , ресурсы которого предоставляются другим соединенным с ним компьютерам — рабочим станциям

Локальные сети обычно объединяют компьютеры, размещенные в одном здании, и не позволяют обеспечить совместный доступ к информации пользователям, находящимся, например, в разных частях города. Для решения этой проблемы создаются региональные сети, объединяющие компьютеры в пределах одного региона (города, страны, континента)

Многие организации, заинтересованные в защите информации от несанкционированного доступа, создают собственные, так называемые **корпоративные сети**. Такие сети могут объединять десятки тысяч компьютеров, размещенных в различных городах и странах.

Глобальные сети

Глобальная сеть — соединения локальных, региональных и корпоративных сетей и отдельных компьютеров, находящихся на больших расстояниях друг от друга.

Большие расстояния требуют наличия дополнительного устройства для обработки больших объемов информации и пересылки ее на большие расстояния. Это — **серверы** глобальной сети, представляющие собой очень мощные компьютеры.

Для пользования глобальной сетью компьютер должен иметь модем и **специальное программное обеспечение**.

Интернет — всемирная глобальная компьютерная сеть, объединяющая в единое целое десятки тысяч разнородных локальных, региональных и корпоративных сетей

Различные варианты подключения к глобальной компьютерной сети Интернет



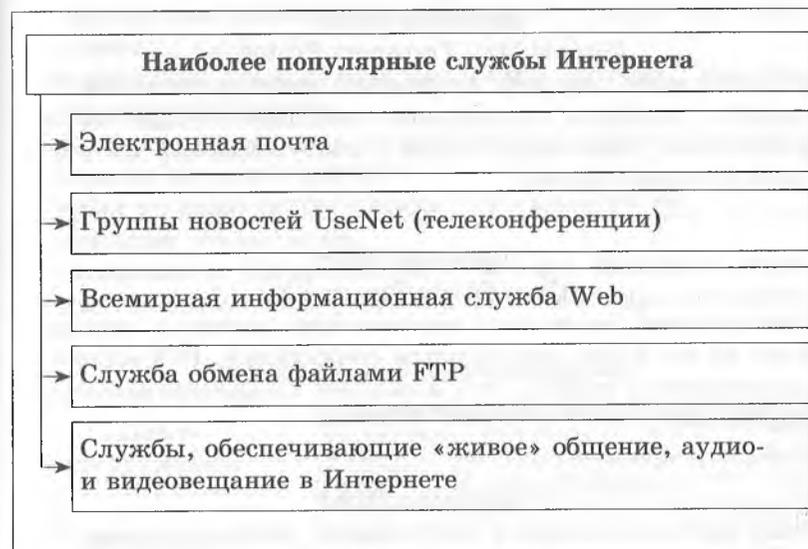
Адресация в сети Интернет

Основные типы адресации в сети	
IP-адрес	Последовательность из четырех чисел, разделенных точками. Каждое из чисел занимает 1 байт, т. е. может принимать значение от 0 до 255. IP-адрес включает в себя номер сети и номер компьютера. Например: 105.65.178.5
URL-адрес	Адрес соответствующей страницы в сети Интернет, сайта или совокупности страниц. URL-адрес состоит из имени протокола или сервиса, который необходимо использовать для доступа к ресурсу, доменного имени ресурса, пути и имени файла, в котором сохраняется документ. URL-адрес содержит латинские буквы и некоторые символы (_ , ~ , —) — между ними не должно быть пробелов. Прописные и строчные буквы различаются. Например: http://home.microsoft.com/intl/ru/tour.html
Доменный адрес	Уникальное символьное имя компьютера в сети. Оно состоит из частей, разделенных точками. Место назначения уточняется справа налево: сначала указывается имя компьютера, затем — имя в сети, в которой он находится. Например: mon.gov.ru

Некоторые имена доменов верхнего уровня

Административные	Тип организации	Географические	Страна
com, biz	Коммерческая	ru	Россия
edu	Образовательная	de	Германия
net	Коммуникационная	it	Италия
org, pro	Некоммерческая	jp	Япония

2.8. УСЛУГИ КОМПЬЮТЕРНЫХ СЕТЕЙ: ЭЛЕКТРОННАЯ ПОЧТА, ТЕЛЕКОНФЕРЕНЦИЯ, WORLD WIDE WEB



Электронная почта

Электронная почта — система пересылки электронной корреспонденции между пользователями телекоммуникационной сети. Ее достоинства — дешевизна, оперативность, доступность.

Почтовый адрес в Интернете состоит из двух частей, разделенных символом @

Записывается почтовый адрес только латинскими буквами; запись не должна содержать пробелов.

$$\text{user_name@mail.ru}$$
 имя пользователя имя сервера

Работа E-mail основана на специальных протоколах, а также на взаимодействии двух программ — почтового сервера и почтового клиента.

Протоколы, по которым работает электронная почта

Протокол SMTP (Simple Mail Transport Protocol)

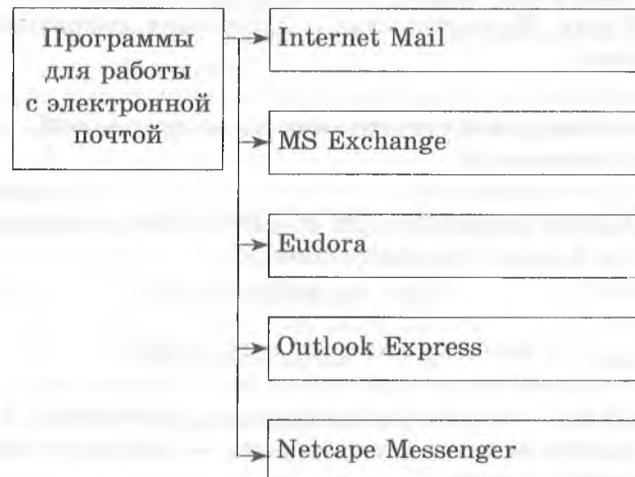
Поддерживает передачу сообщений между адресатами Интернета, позволяет группировать сообщения в адрес одного получателя, размножать копии E-mail-сообщений для передачи на разные адреса.

Протокол POP (Post Office Protocol 3)

Обеспечивает конечному пользователю доступ к пришедшим на его адрес электронным сообщениям. POP-клиенты запрашивают пароль, что повышает конфиденциальность переписки.

Протокол IMAP

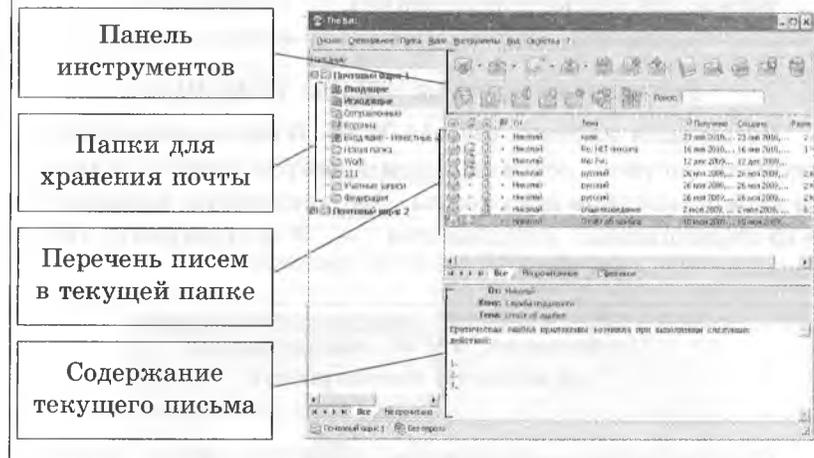
Этот протокол получает все большее распространение. Он дает возможность пользователю работать с письмами непосредственно на сервере провайдера и экономить время работы в сети Интернет.



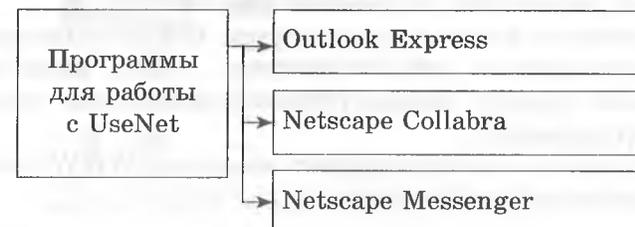
Основные функции почтовых программ:

- извлечение писем из почтового ящика на сервере;
- отправка подготовленных писем;
- дешифровка сообщений, полученных в различных кодировках;
- управление приоритетом сообщений (срочное, обычное, по дешевому тарифу);
- сортировка принятых и отправляемых писем по папкам;
- ведение адресной книги;
- ответ на приходящие письма, пересылка писем;
- создание новых писем;
- возможность прикрепления к письму, имеющему обычный текстовый вид, картинки или любого другого файла.

Интерфейс почтовой программы



Телеконференция (группа новостей)



Программы для работы с конференциями обеспечивают:

- указание множества конференций, в работе которых намерен участвовать пользователь. Эта операция называется *подпиской*, а множество конференций, на которые осуществлена подписка, — *списком подписки* (он может быть в любой момент изменен);
- просмотр фамилий авторов и заголовков (тем) статей в любой конкретной конференции из списка подписки;
- просмотр содержимого статей и при необходимости сохранение их в файле в заданном каталоге компьютера пользователя;
- публикацию своей собственной статьи;
- персональный ответ автору какой-либо статьи на его E-mail-адрес;
- публичный ответ автору конкретной статьи (будет значиться как статья конференции).

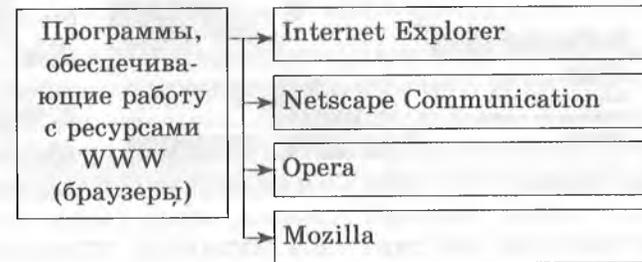
Всемирная паутина (World Wide Web)

Всемирная паутина (World Wide Web) — это информационная система, основным компонентом которой являются гипертекстовые документы. Эти документы называются *Web-страницами*, компьютеры — *Web-серверами* (Web-узлами, Web-сайтами).

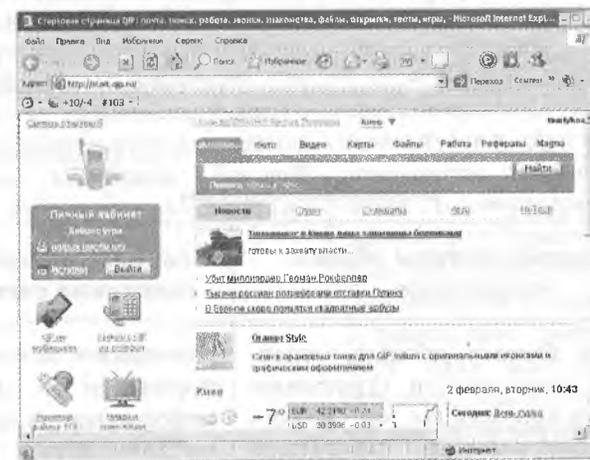
Особенности WWW, обеспечившие ее высокую популярность

- Гипертекстовая организация информационных элементов — страниц WWW. Это означает, что страница WWW может содержать *гипертекстовые ссылки*, оформленные в виде рисунка или выделенного текста.
- Возможность включения в страницы WWW современных мультимедийных средств (графики, звука, анимации), а также средств художественного оформления страниц (макетирования).
- Возможность передачи на сайт владельца WWW-страницы различной информации.

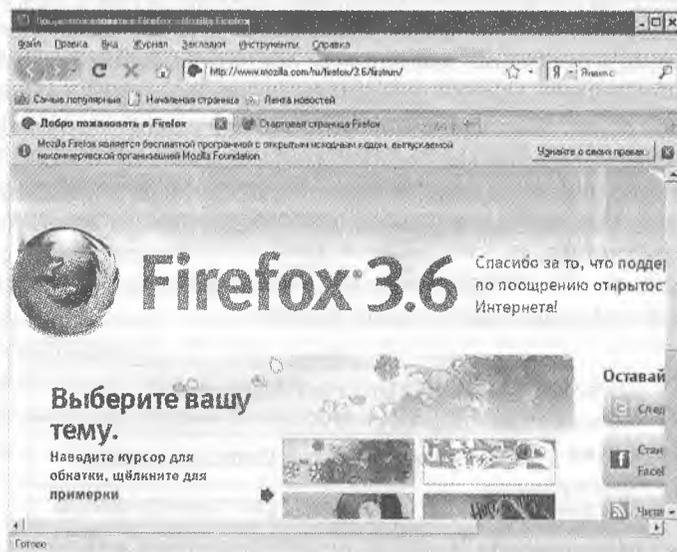
- Наличие бесплатного, хорошего и достаточно простого программного обеспечения, позволяющего непрофессиональному пользователю не только просматривать, но и самому создавать WWW-страницы.
- Наличие хороших поисковых систем и удобных средств запоминания адресов размещения необходимой информации.
- Возможность быстрого перемещения вперед-назад по уже просмотренным страницам, отсутствие необходимости знать IP-адреса хостов, на которых размещена нужная информация.
- Наличие средств обеспечения надежности и конфиденциальности информационного обмена.



Браузер Internet Explorer



Браузер Mozilla Firefox



2.9. ПОИСК ИНФОРМАЦИИ В ИНТЕРНЕТЕ

Для поиска информации в Интернете используются специальные **поисковые серверы**, которые содержат постоянно обновляемую информацию о Web-страницах, файлах и других документах, хранящихся на серверах Интернета.

Поисковые серверы	
Поисковые системы общего назначения	Специализированные поисковые системы
Являются базами данных, содержащими тематически сгруппированную информацию об информационных ресурсах Всемирной	Позволяют искать информацию в других информационных слоях Интернета:

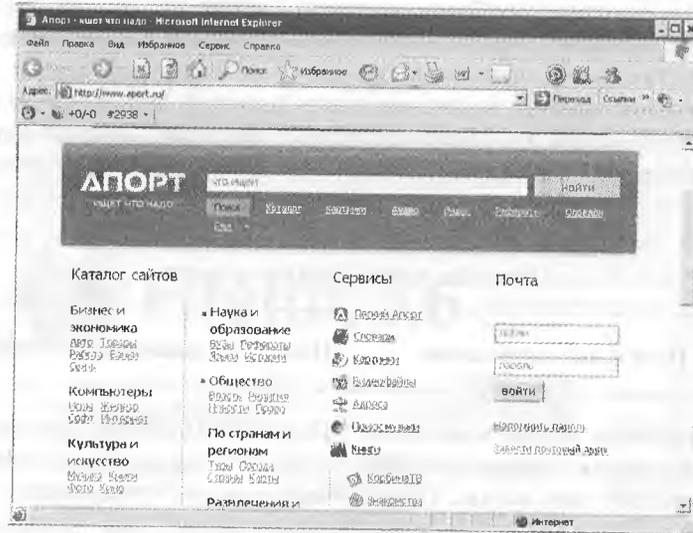
паутины. Заполнение баз данных осуществляется с помощью специальных программ-роботов, которые периодически «обходят» Web-серверы Интернета, читают встречающиеся документы, выделяют в них ключевые слова и заносят в базу, содержащую URL-адреса документов

серверах файловых архивов, почтовых серверах и др.; позволяют также найти адрес электронной почты по имени человека или, наоборот, имя по адресу электронной почты

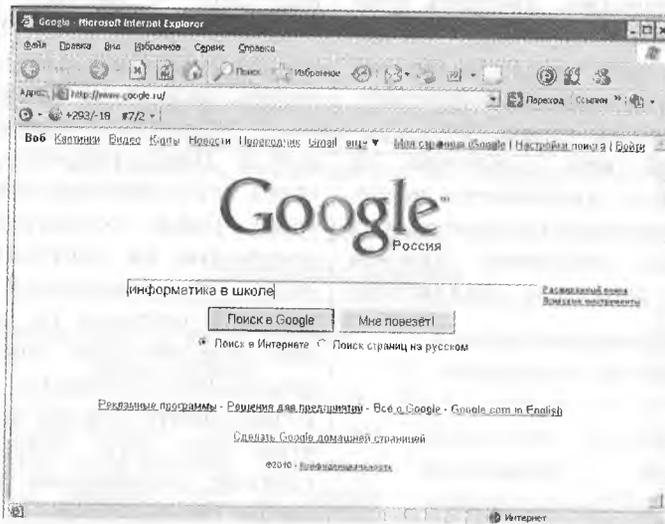
Способы организации поиска

Поиск по каталогам	Поиск по ключевым словам
<p><i>Каталоги</i> в WWW аналогичны систематизированным библиотечным каталогам. Поиск по каталогам состоит в последовательном движении по иерархическому списку ссылок, называемых <i>рубриками</i>, или <i>категориями</i>. Двигаясь вниз по детализирующим категориям, можно найти страницы с нужной информацией. На каждой странице, открываемой при движении по каталогу, указывается последовательность просмотренных рубрик, например: <i>Деловой мир: Финансы: Аналитика</i> и т. д.</p>	<p>Поисковая система дает пользователю возможность сформулировать требования к необходимой ему информации — создать <i>запрос</i>. Простой запрос содержит одно или несколько ключевых слов — главных для искомого документа. Можно создать также сложный запрос, используя логические операции, шаблоны и т. д. После отправки запроса поисковая машина автоматически просматривает документы на контролируемых (<i>индексируемых</i>) ею сайтах и отбирает те, которые соответствуют запросу (<i>релевантны</i> запросу). В результате поиска создается одна или несколько страниц, содержащих ссылки на релевантные запросу документы</p>

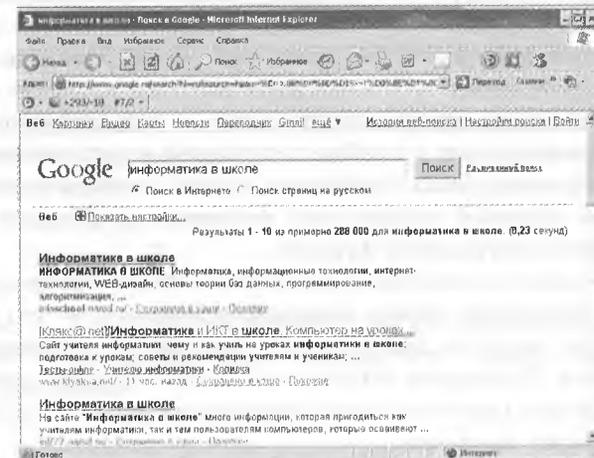
Тематические каталоги поисковой системы Апорт



Поиск по ключевым словам в системе Google



Результат поиска по ключевым словам



2.10. ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Чтение короткой (30–50 строк) простой программы на алгоритмическом языке (языке программирования)

При решении любой задачи на компьютере предполагается, что некоторая информация подвергается обработке по предварительно составленной инструкции, называемой программой.

Программа — это алгоритм, записанный на языке программирования.

Язык, на котором записывается алгоритм выполнения программы, называется языком программирования.

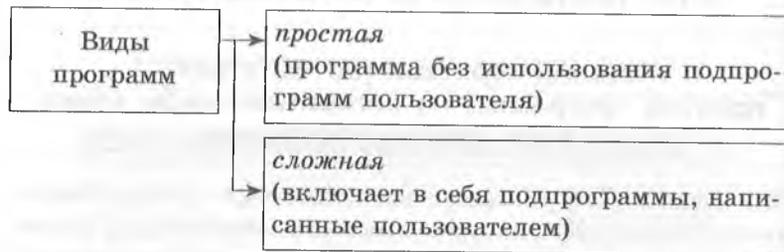
Для записи программ используется конечный набор символов, составляющих алфавит языка программирования. Алфавит языка программирования включает в себя, кроме букв, также цифры, знаки препинания, знаки арифметических действий и некоторые другие дополнительные символы. Программа записывается в виде последовательности символов из алфавита своего языка программирования. Естественно,

что не любой текст, составленный из символов алфавита, будет являться правильной программой. Правильность построения программы из символов алфавита можно проверить, используя синтаксис языка программирования.

Синтаксис языка программирования — это набор правил, которые определяют способы построения программ из символов алфавита. Зная синтаксис языка, можно построить алгоритм, который определяет, является ли данный текст правильной программой или нет.

Должна быть определена и семантика языка программирования. **Семантика языка программирования** — это набор правил, по которым исполнитель выполняет программы на этом языке. Пользуясь семантикой языка, можно однозначно определить результат выполнения программы с заданными входными данными.

При чтении программы необходимо сначала определить, к какому виду она относится.



Чтение программ

Простая программа	Сложная программа
<p>Необходимо выяснить:</p> <ul style="list-style-type: none"> • что является входными данными и как они вводятся в программу; • какие действия последовательно выполняются с помощью каждого функционального узла программы (операторов), т. е. рассмотреть 	<p>Необходимо:</p> <ul style="list-style-type: none"> • проанализировать, что и как выполняют подпрограммы, каковы их входные и выходные параметры;

- пошаговое выполнение операторов, обратив при этом внимание на роль вспомогательных переменных, массивов и т. д.;
- что является результатами работы программы;
- какие есть ограничения по работе алгоритма

- в основной программе вызовы каждой из подпрограмм рассматривать как результат работы соответствующего подалгоритма

Пример 1. Даны записи одной и той же программы на двух языках программирования. Определить, какую задачу решает программа.

Pascal

```

Program Pr_1;
Var
  i, N, M, K, NOD: integer;
Begin
  Repeat
    Write('Введите натуральное число N = ');
    Readln(N);
    Write('Введите натуральное число M = ');
    Readln(M);
  Until (N > 0) and (M > 0);
  If N > M then K := M else K = N;
  NOD := 1;
  For i := 2 to K do
  Begin
    If (N mod i = 0) and (M mod i = 0) then
      Begin
        NOD := i;
      End;
  End;
  Writeln('NOD = ', NOD);
End.

```

QBASIC

```

REM PR_1
INPUT "N = "; N
INPUT "M = "; M
WHILE (N <= 0) OR (M <= 0)
  INPUT "N = "; N
  INPUT "M="; M
WEND
IF N > M THEN K = M ELSE K = N
NOD = 1
FOR I = 2 TO K
  IF (N\I = N/I) AND (M\I = M/I) THEN NOD = I
NEXT I
PRINT "MOD("; N; ", "; M; ") = "; NOD

```

Решение. Проанализируем тексты программ:

- 1) формируется тело программы и описываются переменные;
- 2) вводятся натуральные числа M и N , причем проверяется условие корректности ввода: числа должны быть положительными. Если введенные значения не удовлетворяют условию, то ввод повторяют, пока условие не будет выполнено;
- 3) выбирается наименьшее значение из M и N , результат записывается в K ;
- 4) NOD присваивается значение 1;
- 5) в цикле от 2 до K генерируется число I ;
- 6) тело цикла: в условном операторе проверяется, является ли значение переменной I одновременно делителем M и N . Если условие выполняется, то текущее значение I сохраняется в переменной NOD ; если условие не выполняется, NOD не изменит своего значения;
- 7) после перебора всех значений I в NOD или запишется наибольший делитель двух чисел M и N , или останется значение 1;
- 8) последний оператор программы служит для вывода результата работы программы — значения переменной NOD .

Переменные, используемые в программе:

N, M — исследуемые числа;

I — переменная цикла;

NOD — наибольший общий делитель;

K — наименьшее из M и N .

Ответ: данная программа позволяет определить для двух чисел M и N их наибольший общий делитель NOD .

Примечание. Эту же задачу можно решить, используя алгоритм Евклида.

Пример 2. Дана программа на двух языках программирования. Определить, какую задачу она решает.

Pascal

```

Program Pr_2;
Const M = 100;
Var
  MAS: array[1..M] of integer;
  i, Max, K, N: integer;
Begin
  Write('Введите размер массива N = ');
  Readln(N);
  Write('Введите первый элемент массива ');
  Readln(MAS[1]);
  MAX := MAS[1]; K := 1;
  For i := 2 to N do
    Begin
      Write('Введите элемент массива ');
      Readln(MAS[i]);
      If MAX <= MAS[i] then
        Begin
          MAX := MAS[i]; K:= i;
        End;
    End;
  Writeln('MAX =', MAX, 'номер элемента ', K);
End.

```

QBASIC

```

REM PR_2
'нумерация элементов массива начинается с 1
OPTIM BASE 1
DO
  INPUT "Введите размер массива N = "; N%
LOOP UNTIL N% >= 0
  DIM MAS (N%)
  INPUT "MAS(1) = "; MAS(1)
  K = 1
  FOR I = 2 TO N%
    PRINT "MAS(" ; I ; ") = "
    INPUT MAS(I)
    IF MAX <= MAS(I) THEN
      K = I
      MAX = MAS(I)
    ENDIF
  NEXT I
PRINT "MAX = "; MAX; "K = "; K

```

Решение. Проанализируем тексты программы:

- 1) формируется тело программы, описываются переменные и одномерный массив *MAS* целого типа (для Pascal целый массив длиной 100);
- 2) вводится фактическая длина массива *N* с проверкой на положительное значение *N*;
- 3) вводится значение первого элемента массива *MAS*;
- 4) устанавливается начальное значение *MAX* по первому элементу массива;
- 5) переменной *K* присваивается значение 1;
- 6) последовательно, в цикле, просматриваются вводимые элементы массива, и если очередной элемент *MAS(I)* больше или равен *MAX*, то переписывается значение *MAS(I)* в *MAX* и в переменной *K* запоминается *I*;
- 7) выводятся результаты: *MAX* — значение максимального элемента массива и *K* — номер максимального элемента в исходном массиве (если таких элементов несколько, выведется номер самого правого максимума).

Переменные, используемые в программе:

MAS — массив чисел;

N — размер массива;

I — переменная цикла;

MAX — значение наибольшего элемента;

K — номер наибольшего элемента.

Ответ: программа написана для поиска значения максимального элемента массива и его номера (если таких элементов несколько, то будет найден номер самого правого максимума).

Поиск и исправление ошибок в небольшом фрагменте программы (10–20 строк)

Аспекты проверки программы

Правильность

→ Экспериментальная проверка правильности программы: в точности ли она выполняет то, для чего предназначена

Эффективность реализации

→ Отыскание способов «заставить» правильную программу работать быстрее или расходовать меньше памяти

Сложность вычислений

→ Экспериментальный анализ сложности алгоритма или экспериментальное сравнение двух или более алгоритмов, решающих одну и ту же задачу

Методы теоретического контроля текста программы

Просмотр

Текст программы просматривается с целью обнаружения явных ошибок и расхождений с алгоритмом. Следует просмотреть организацию всех циклов, чтобы убедиться в правильности операторов, задающих кратности циклов. Полезно еще раз просмотреть условия в условных операторах, аргументы в обращениях к подпрограммам и т. д.

Проверка

При проверке программы программист по ее тексту мысленно старается восстановить вычислительный процесс, определяемый программой, после чего сверяет этот текст с требуемым процессом

Прокрутка

Основой этого метода является имитация программистом за столом выполнения программы на машине. Для выполнения прокрутки приходится задаваться какими-либо исходными данными и производить над ними необходимые вычисления. Прокрутка — трудоемкий процесс, поэтому ее следует применять лишь для контроля логически сложных участков программы

Отладка и тестирование программы на компьютере

Отладка — это процесс поиска и устранения ошибок в программе, производимый по результатам ее прогона на компьютере.

При отладке происходит локализация и устранение синтаксических ошибок и явных ошибок кодирования

Тестирование — это испытание, проверка правильности работы программы в целом или ее составных частей.

В процессе тестирования проверяется работоспособность программы, не содержащей явных ошибок

Тестирование устанавливает наличие каких-либо ошибок, а отладка выясняет их причину.

Как бы тщательно ни была отлажена программа, решающим этапом, устанавливающим ее пригодность для работы, является контроль программы по результатам ее выполнения на основе **системы тестов**. Программу условно можно считать правильной, если ее запуск для выбранной системы тестовых исходных данных всегда дает правильные результаты. Тестовые данные должны обеспечить проверку всех возможных условий возникновения ошибок.

Процесс тестирования можно разделить на три этапа:

1. Проверка в нормальных условиях. Предполагает тестирование на основе данных, которые характерны для реальных условий функционирования программы.
2. Проверка в экстремальных условиях. Тестовые данные включают в себя граничные значения области изменения входных переменных, которые должны восприниматься программой как правильные данные. Типичными примерами таких значений являются очень маленькие или очень большие числа, а также отсутствие данных. Еще один тип экстремальных условий — это граничные объемы данных, например: массивы состоят из слишком малого или слишком большого числа элементов.

3. Проверка в исключительных ситуациях. Она проводится с использованием данных, значения которых лежат за пределами допустимой области изменений.

Примечание: ошибки могут быть допущены на всех этапах решения задачи.

Разновидности характерных ошибок

- *Неправильная постановка задачи* — верное решение неверно сформулированной задачи.
- *Неверный алгоритм* — выбор алгоритма, приводящего к неточному, неэффективному решению задачи.
- *Ошибки анализа* — неполный учет ситуаций, которые могут возникнуть; логические ошибки.
- *Семантические ошибки* — неправильный порядок выполнения оператора.
- *Синтаксические ошибки* — нарушение правил, определяемых языком программирования.
- *Ошибки при выполнении операций* — слишком большое число, переполнение, деление на нуль, извлечение квадратного корня из отрицательного числа и т. д.
- *Ошибки в данных* — неправильное определение возможного диапазона изменения данных.
- *Опечатки* — перепутаны близкие по написанию символы, например цифра 1 и буква I.
- *Ошибки ввода-вывода* — неверное считывание входных данных, неверное задание форматов, отсутствие некоторых данных.

Создание собственной программы (30–50 строк) для решения простых задач

При написании программы следует прежде всего четко уяснить задачу, которую должна решать программа. Затем предварительно разработанный алгоритм решения задачи записывается в виде упорядоченной последовательности команд (инструкций), т. е. составляется программа, ориентированная на определенную среду программирования.

При написании программы необходимо обязательно проверять, насколько эта программа соответствует намеченной цели, т. е. делает ли она для всех наборов данных то, что от нее требуется, не выполняет ли она каких-то лишних действий. Основное внимание следует уделить предупреждению логических ошибок. Для этого рекомендуется перед написанием программы построить блок-схему алгоритма решения, которая позволяет абстрагироваться от конкретного языка программирования и сосредоточиться на анализе алгоритма.

Пример 1. Составить словесный алгоритм, разработать блок-схему и написать программу проверки принадлежности введенного числа данной арифметической прогрессии. Прогрессия задается двумя последовательными членами.

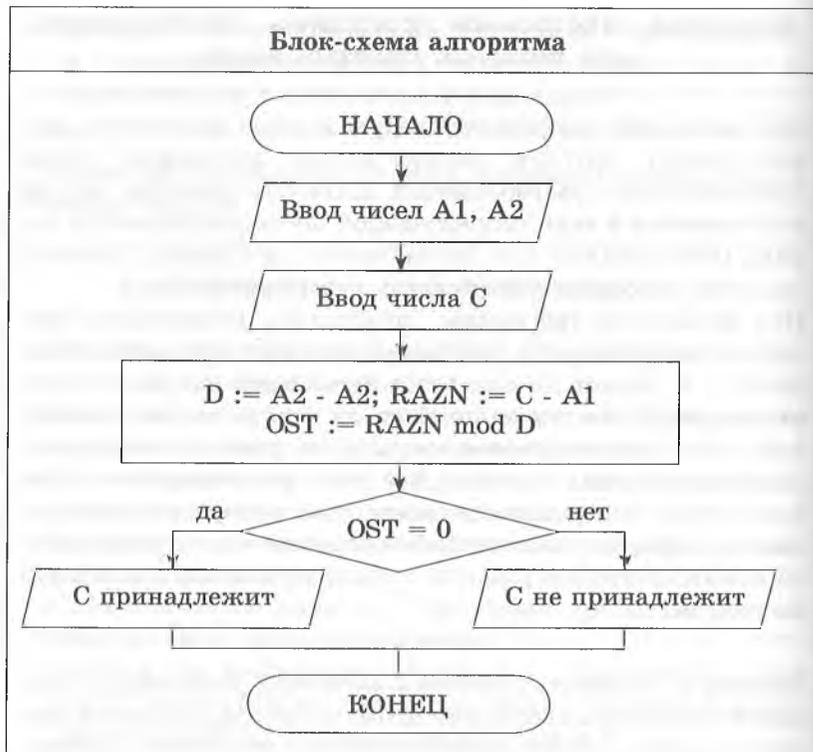
Решение.

Словесный алгоритм

Начало алгоритма:

1. Ввести два последовательных члена арифметической прогрессии: A_1, A_2 .
2. Ввести произвольное целое число C .
3. Найти разность (D) арифметической прогрессии.
4. Найти разность между введенным числом C и членом арифметической прогрессии, например A_1 .
5. Найти остаток от деления нацело найденной разности на D .
6. Если остаток от деления равен 0, значит, число C принадлежит рассматриваемой арифметической прогрессии; иначе получаем, что число C не принадлежит рассматриваемой арифметической прогрессии.

Конец алгоритма



Программы

Pascal

```

Program Progress;
Var
  A1, A2, C, D, RAZN, OST : integer;
Begin
  Writeln('Введите два последовательных члена арифметической прогрессии');
  Readln(A1, A2);
  Writeln('Введите целое число');
  Readln(C);
  D := A2 - A1;
  RAZN := C - A1;
  OST := RAZN mod D;

```

```

If OST = 0 then
  Writeln('Число', C, ' принадлежит прогрессии')
else
  Writeln('Число', C, ' не принадлежит прогрессии');
End.

```

QBASIC

```

REM PROGRESS
PRINT "Введите два последовательных члена арифметической прогрессии"
INPUT A1, A2
PRINT "Введите число"
INPUT C
D = A2 - A1
RAZN = C - A1
OST = RAZN MOD D
IF (OST=0) THEN
  PRINT "Число "; C; " принадлежит прогрессии"
ELSE
  PRINT "Число "; C; " не принадлежит прогрессии"
ENDIF
END

```

Пример 2. Составить словесный алгоритм, алгоритм в виде блок-схемы и написать программу поиска в строковом массиве, содержащем фамилии 10 учеников, заданной фамилии, обеспечить запоминание ее порядкового номера (массив фамилий может быть не упорядочен).

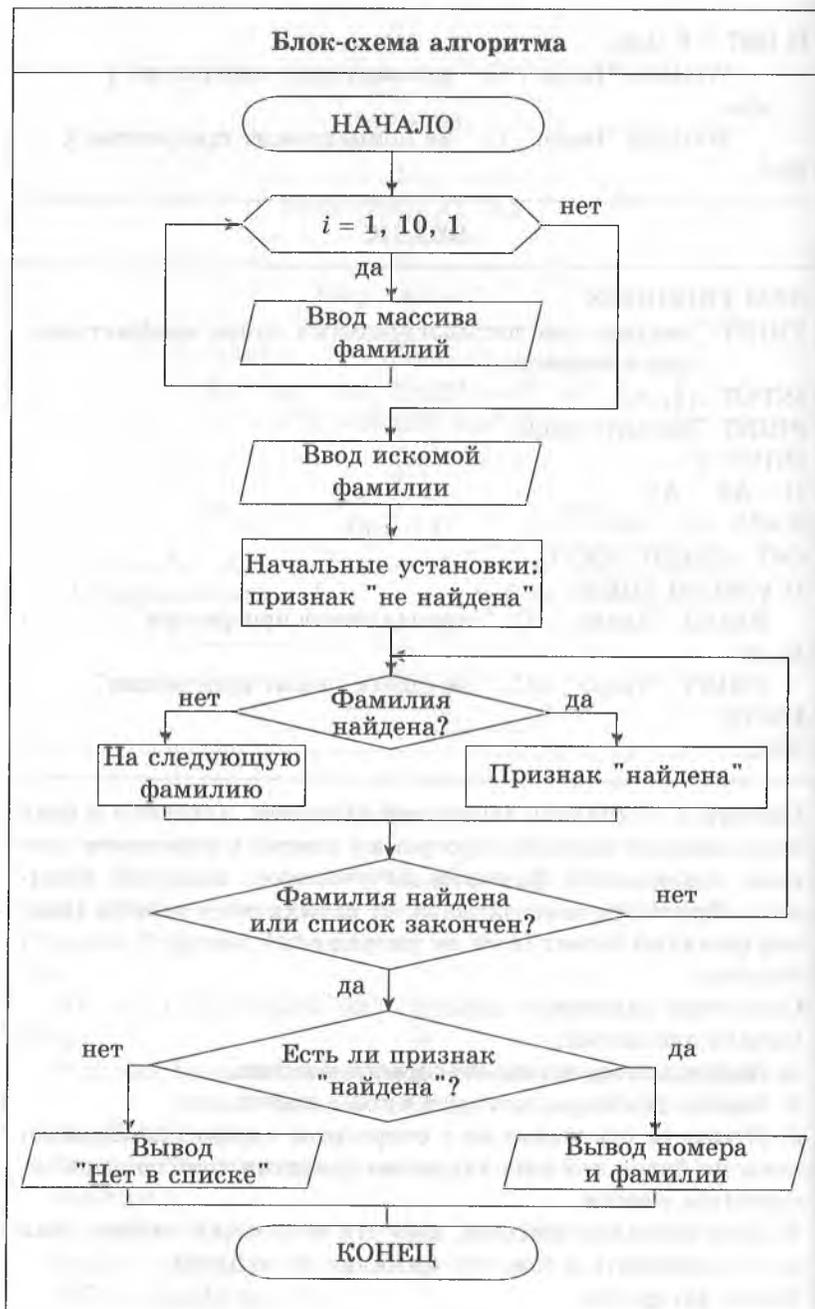
Решение.

Словесный алгоритм

Начало алгоритма:

1. Ввести все элементы строкового массива.
2. Ввести фамилию, которую нужно найти.
3. Провести сравнение ее с очередным элементом массива, пока не будет найдена такая же фамилия или пока не закончится список.
4. Если фамилия найдена, вывести ее номер в списке, если нет — сообщить о том, что фамилия не найдена.

Конец алгоритма

**Программы****Pascal**

```

Program MAS;
Var
  Fam: array[1..10] of string;
  i: byte;
  STR: string;
  f: boolean;
Begin
  Writeln('Введите массив фамилий');
  For i := 1 to 10 do Readln(Fam[i]);
  Writeln('Введите фамилию для поиска');
  Readln(STR);
  f := false;
  i := 1;
  Repeat
    If Fam[i] = STR then
      f := true
    else
      i := i + 1;
  Until f or (i > 10);
  If f then Writeln('Фамилия находится на', i, '-м месте')
  else Writeln('Фамилия не найдена');
End.
  
```

QBASIC

```

REM MAS
'нумерация элементов массива начинается с 1
OPTIN BASE 1
DIM FAM$(10)
PRINT "Введите массив фамилий"
FOR I := 1 TO 10 INPUT FAM$(I)
PRINT "Введите фамилию для поиска"
INPUT STR$
  
```

```
F = 0
I = 1
DO
  IF FAM$(I) = STR$ THEN F = 1
  ELSE NEXT I
ENDIF
LOOP UNTIL (F = 1) OR (I > 10)
IF F = 1 THEN
  PRINT "Фамилия находится на ", I, "-м месте"
ELSE
  PRINT "Фамилия не найдена"
ENDIF
```

Справочное издание

Для старшего школьного возраста

НАГЛЯДНО И ДОСТУПНО

**Трофимова Ирина Алексеевна
Яровая Ольга Владимировна**

ИНФОРМАТИКА В СХЕМАХ И ТАБЛИЦАХ

Ответственный редактор *А. Жилинская*
Ведущий редактор *Т. Судакова*
Редактор *О. Каширина*
Художественный редактор *Е. Брынчик*

ООО «Издательство «Эксмо»
127299, Москва, ул. Клары Цеткин, д. 18/5. Тел. 411-68-86, 956-39-21.
Home page: www.eksmo.ru E-mail: info@eksmo.ru

Подписано в печать 25.10.2011.
Формат 60x90^{1/16}. Печать офсетная. Усл. печ. л. 10,0.
Тираж 3000 экз. Заказ № 533.

Отпечатано в ГП ПО «Псковская областная типография».
180004, г. Псков, ул. Ротная, 34.

ISBN 978-5-699-35281-4



9 785699 352814 >

Оптовая торговля книгами «ЭКСМО»:
ООО «ТД «Эксмо». 142700, Московская обл., Ленинский р-н, г. Видное,
Белокаменное ш., д. 1, многоканальный тел. 411-50-74.
E-mail: reception@eksmo-sale.ru

По вопросам приобретения книг «Эксмо» зарубежными оптовыми покупателями обращаться в отдел зарубежных продаж ТД «Эксмо»
E-mail: international@eksmo-sale.ru

International Sales: International wholesale customers should contact
Foreign Sales Department of Trading House «Eksmo» for their orders.
international@eksmo-sale.ru

По вопросам заказа книг корпоративным клиентам,
в том числе в специальном оформлении,
обращаться по тел. 411-68-59, доб. 2115, 2117, 2118, 411-68-99, доб. 2762, 1234.
E-mail: vipzakaz@eksmo.ru

Оптовая торговля бумажно-беловыми и канцелярскими товарами для школы и офиса «Канц-Эксмо»:
Компания «Канц-Эксмо»: 142702, Московская обл., Ленинский р-н, г. Видное-2,
Белокаменное ш., д. 1, а/я 5. Тел./факс +7 (495) 745-28-87 (многоканальный).
e-mail: kanc@eksmo-sale.ru, сайт: www.kanc-eksmo.ru

Полный ассортимент книг издательства «Эксмо» для оптовых покупателей:

В Санкт-Петербурге: ООО СЗКО, пр-т Обуховской Обороны, д. 84Е.
Тел. (812) 365-46-03/04.

В Нижнем Новгороде: ООО ТД «Эксмо НН», ул. Маршала Воронова, д. 3.
Тел. (8312) 72-36-70.

В Казани: Филиал ООО «РДЦ-Самара», ул. Фрезерная, д. 5.
Тел. (843) 570-40-45/46.

В Ростове-на-Дону: ООО «РДЦ-Ростов», пр. Стачки, 243А.
Тел. (863) 220-19-34.

В Самаре: ООО «РДЦ-Самара», пр-т Кирова, д. 75/1, литера «Е».
Тел. (846) 269-66-70.

В Екатеринбурге: ООО «РДЦ-Екатеринбург», ул. Прибалтийская, д. 24а.
Тел. +7 (343) 272-72-01/02/03/04/05/06/07/08.

В Новосибирске: ООО «РДЦ-Новосибирск», Комбинатский пер., д. 3.
Тел. +7 (383) 289-91-42. E-mail: eksmo-nsk@yandex.ru

В Киеве: ООО «РДЦ Эксмо-Украина», Московский пр-т, д. 9.
Тел./факс: (044) 495-79-80/81.

Во Львове: ТП ООО «Эксмо-Запад», ул. Буэкова, д. 2.
Тел./факс (032) 245-00-19.

В Симферополе: ООО «Эксмо-Крым», ул. Киевская, д. 153.
Тел./факс (0652) 22-90-03, 54-32-99.

В Казахстане: ТОО «РДЦ-Алматы», ул. Домбровского, д. 3а.
Тел./факс (727) 251-59-90/91. rdc-almaty@mail.ru

Полный ассортимент продукции издательства «Эксмо»
можно приобрести в магазинах «Новый книжный» и «Читай-город».
Телефон единой справочной: 8 (800) 444-8-444.
Звонок по России бесплатный.

В Санкт-Петербурге в сети магазинов «Буквоед»:
«Парк культуры и чтения», Невский пр-т, д. 46. Тел. (812) 601-0-601
www.bookvoed.ru

По вопросам размещения рекламы в книгах издательства «Эксмо»
обращаться в рекламный отдел. Тел. 411-68-74.