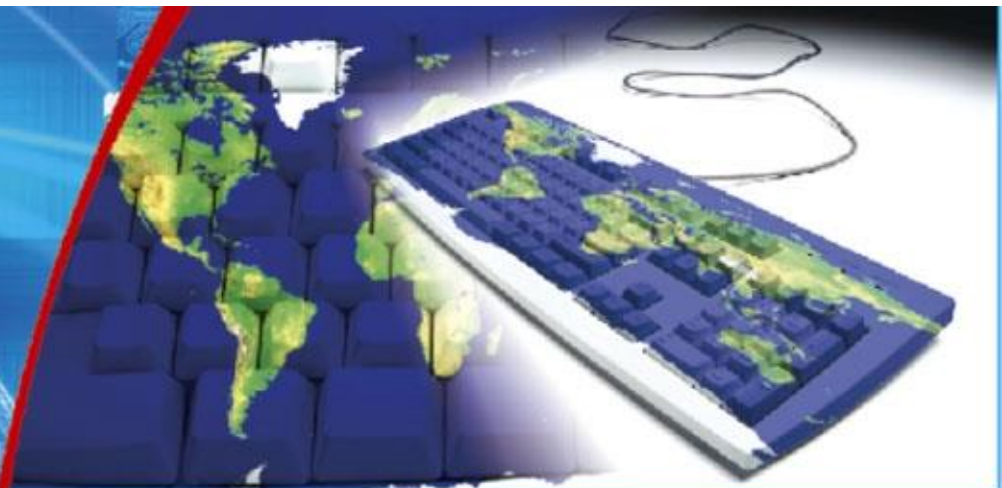


ИНФОРМАТИКА



АТИКА • ИНФОРМАТИКА • ИНФОРМАТИКА • ИНФОРМАТИКА • ИНФОРМАТИКА • ИНФОРМАТИКА •



КУРС ЛЕКЦИЙ
по направлению подготовки бакалавров
очной формы обучения

БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ

Кафедра математики, физики и информатики

Петракова Наталья Васильевна

ИНФОРМАТИКА

КУРС ЛЕЦИЙ

ПО НАПРАВЛЕНИЮ ПОДГОТОВКИ БАКАЛАВРОВ

ОЧНОЙ ФОРМЫ ОБУЧЕНИЯ

БРЯНСКАЯ ОБЛАСТЬ
2015

УДК 002.5/.6(07)

ББК 32.81

П. 29

Петракова Н.В. Информатика. КУРС ЛЕКЦИЙ: Учебное пособие /Н.В. Петракова. – Брянск. Издательство Брянский ГАУ, 2015. - 172 с.

В учебном пособии изложены базовые понятия по информатике, современным компьютерным аппаратным средствам. Рассмотрены вопросы организации размещения, обработки, хранения и передачи информации. Раскрыты назначение, возможности применения и дана классификация программного обеспечения. Приведены эффективные приемы работы с офисными программами. Излагаются основы теории алгоритмов и технологии программирования. Дается представление о методах и средствах моделирования. Описаны услуги компьютерных сетей.

Для студентов по направлениям подготовки бакалавров очной формы обучения высшего образования: 15.03.04 «Автоматизация технологических процессов и производств», 35.03.06 «Агроинженерия», 13.03.02 «Электроэнергетика и электротехника», 21.03.02 «Землеустройство и кадастры» и 20.03.02 «Природоустройство и водопользование».

Рецензенты: доцент кафедры информационных систем и технологий,

к.э.н. Ульянова Н.Д.

доцент кафедры математики, физики и информатики,

к.т.н. Панкова Е.А.

Рекомендовано к изданию методическим советом факультета энергетики и природопользования от 22.06. 2015 г., протокол №6.

© Н.В. Петракова, 2015

© Брянсий ГАУ, 2015

СОДЕРЖАНИЕ

<i>Лекция 1.</i>	
ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ИНФОРМАТИКИ	7
1. Определение информатики. Понятие информации.	7
2. Информационные процессы.	11
3. Измерение информации.	13
4. Кодирование информации.	16
Контрольные вопросы	29
<i>Лекция 2.</i>	
ЛОГИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ	30
1. Формы мышления.	30
2. Логические операции над высказываниями.	31
3. Логические выражения и таблицы истинности.	35
Контрольные вопросы	38
<i>Лекция 3.</i>	
ТЕХНИЧЕСКИЕ СРЕДСТВА ОБРАБОТКИ ИНФОРМАЦИИ	39
1. Классическая архитектура ЭВМ.	39
2. Магистрально-модульный принцип построения персонального компьютера.	44
3. Базовая аппаратная конфигурация персонального компьютера.	48
4. Периферийные устройства персонального компьютера.	51
Контрольные вопросы	52
<i>Лекция 4.</i>	
КЛАССИФИКАЦИЯ И ТЕНДЕНЦИИ РАЗВИТИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	53
1. Виды программного обеспечения.	53
2. Установка программ.	57
3. Правовая охрана программ и данных.	57
4. Перспективы развития программного обеспечения.	61
Контрольные вопросы	63
<i>Лекция 5.</i>	
ОПЕРАЦИОННЫЕ СИСТЕМЫ СЕМЕЙСТВА WINDOWS	64
1. Особенности операционных систем семейства Windows.	64
2. Графический интерфейс.	65
3. Файловая система.	66
Контрольные вопросы	70
<i>Лекция 6.</i>	
ОФИСНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	71
1. Текстовый процессор OpenOffice.org Writer.	
Технологии обработки текстовой информации.	71
2. Табличный процессор OpenOffice.org Calc.	
Обработка данных средствами электронных таблиц.	81
3. OpenOffice.org Impress.	
Программные технологии создания мультимедиа-презентаций.	87
Контрольные вопросы	99

<i>Лекция 7</i>	100
ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ	100
1. Основы алгоритмизации процессов обработки информации.....	100
2. Основные понятия и классификация языков программирования.....	107
3. Основные объекты языка программирования QBASIC.....	110
4. Примеры программирования на языке QBASIC.....	128
Контрольные вопросы	132
 <i>Лекция 8</i>	
ОСНОВЫ МОДЕЛИРОВАНИЯ	133
1. Моделирование как метод познания.....	133
2. Формы представления моделей. Формализация.....	136
3. Типы информационных моделей.....	140
Контрольные вопросы	145
 <i>Лекция 9</i>	
КОМПЬЮТЕРНЫЕ СЕТИ	146
1. Основные понятия вычислительной сети.....	146
2. Топология вычислительной сети.....	148
3. Сетевые стандарты.....	150
4. Каналы связи.....	151
5. Передача данных в сети.....	153
6. Коммуникационное оборудование.....	154
Контрольные вопросы	157
 ГЛОССАРИЙ	158
 ЛИТЕРАТУРА	171

Лекция 1.

ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ИНФОРМАТИКИ

1. *Определение информатики. Понятие информации.*
2. *Информационные процессы.*
3. *Измерение информации.*
4. *Кодирование информации.*

1. Определение информатики. Понятие информации.

Слово «**информатика**» в современном значении образовано в результате объединения двух слов: «**информация**» и «**автоматика**». Таким образом, получается «**автоматическая работа с информацией**».

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

Современная информатика включает следующие научные направления:

- **теоретическую информатику** (теорию информации, теорию кодирования, математическую логику, теорию систем и др.);
- **кибернетику** (теорию управления в природе, технике и обществе);
- **искусственный интеллект** (распознавание образов, понимание речи, машинный перевод, логические выводы, алгоритмы самообучения);
- **вычислительную технику** (устройство компьютеров и компьютерных сетей);
- **программирование** (методы создания новых программ);
- **прикладную информатику** (персональные компьютеры, прикладные программы, информационные системы и т.д.).

Информатика в широком смысле представляет собой единство разнообразных отраслей науки, техники и производства, связанных с переработкой ин-

формации главным образом с помощью компьютеров и телекоммуникационных средств связи во всех сферах человеческой деятельности.

Информатика состоит из трех взаимосвязанных частей – *технических средств* (hardware), *программных средств* (software), *алгоритмических средств* (brainware). В свою очередь, информатику, как в целом, так и каждую ее часть обычно рассматривают с разных позиций (рис. 1): как отрасль народного хозяйства, как фундаментальную науку, как прикладную дисциплину.



Рис. 1. Структура информатики как отрасли, науки, прикладной дисциплины

Информатика как *отрасль народного хозяйства* состоит из однородной совокупности предприятий разных форм хозяйствования, где занимаются производством компьютерной техники, программных продуктов и разработкой современной технологии переработки информации.

Информатика как *фундаментальная наука* занимается разработкой методологии создания информационного обеспечения процессов управления любыми объектами на базе компьютерных информационных систем.

Информатика как *прикладная дисциплина* занимается:

- изучением закономерностей в информационных процессах (накопление, переработка, распространение);
- созданием информационных моделей коммуникаций в различных областях человеческой деятельности;
- разработкой информационных систем и технологий в конкретных областях.

Информатика – это дисциплина комплексная, тесно связанная с другими науками, у которых есть общий объект исследования – **информация**.

Понятие **информации** является основополагающим понятием информатики. Любая деятельность человека представляет собой процесс сбора и переработки информации, принятия на ее основе решений и их выполнения.

Термин «**информация**» происходит от латинского **informatio** – *разъяснение, изложение, осведомление*.

Информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают степень неопределенности и неполноты имеющихся о них знаний.

С понятием информации связаны такие понятия, как сигнал, сообщение и данные.

Сигнал (от латинского **signum** – знак) представляет собой любой процесс, несущий информацию.

Сообщение – это информация, представленная в определенной форме и предназначенная для передачи.

Данные – это информация, представленная в формализованном виде и предназначенная для обработки ее техническими средствами.

Все многообразие окружающей нас информации можно сгруппировать по различным признакам, т.е. **классифицировать** по видам. Так, например, в зависимости от **области возникновения** информацию, отражающую процессы

и явления неодушевленной природы называют **элементарной**, процессы животного и растительного мира – **биологической**, человеческого общества – **социальной**.



По способу передачи и восприятия различают следующие виды информации: **визуальную** – передаваемую видимыми образами и символами, **аудиальную** – звуками, **тактильную** – ощущениями, **органолептическую** – запахами и вкусом, **машинную** – выдаваемую и воспринимаемую средствами вычислительной техники.



По форме представления информация бывает – **текстовая**, **числовая**, **графическая**, **звуковая**, **видеоинформация**.



По общественному значению информацию различают – **массовая**, **специальная**, **личная**.



Свойства информации

Объективность и субъективность информации. Та информация, которая отражает явление или объекты материального мира, является **объективной**. Информация, которую создают люди (то есть субъекты), является **субъективной**.

Объективная информация может преобразовываться в субъективную, если в ее обработке или преобразовании участвуют люди. В ходе информационного процесса степень объективности информации всегда понижается.

Полнота информации характеризует качество информации и определяет достаточность данных для принятия решений или создания новых данных на основе имеющихся.

Достоверность информации – отражает реально существующие объекты с необходимой точностью.

Адекватность информации – это степень соответствия реальному состоянию дела.

Доступность информации – мера возможности получить ту или иную информацию.

Актуальность информации – это степень соответствия информации текущему моменту времени.

2. Информационные процессы.

Информационный процесс – это совокупность последовательных действий, производимых над информацией с целью получения результата.

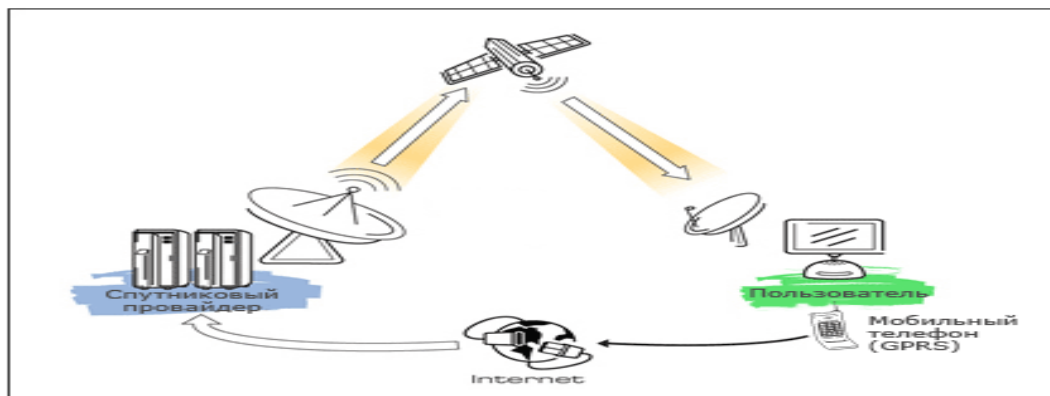
Любая информационная деятельность человека сводится к выполнению основных видов информационных процессов: *передачи и получения, хранения, обработки и использования информации.*

Передача информации.

При **передаче информации** всегда есть два объекта – **источник** и **приемник** информации.



Эти роли могут меняться, например, во время диалога каждый из участников выступает то в роли источника, то в роли приемника информации.



Передача информации может происходить при непосредственном разговоре между людьми, через переписку, с помощью технических средств связи. Такие средства связи называются **каналами передачи информации**. В процессе передачи информация может исказиться или теряться, если информационные каналы плохого качества.

Получение информации – это, прежде всего реализация способности к отражению различных свойств объектов, явлений и процессов в окружающем мире.



Обработка информации – это преобразование информации из одного вида в другой, проводимое по строго формальным правилам.

Хранение информации.

Для хранения информации человек, прежде всего, использует свою память.



Память человека можно условно назвать оперативной. Сохраненные в памяти знания воспроизводятся человеком мгновенно. Свою память мы еще можем назвать *внутренней памятью*. Тогда информацию, сохраненную на внешних носителях (в записных книжках, справочниках, телефонах, магнитных записях), можно назвать нашей *внешней памятью*. Чтобы воспользоваться такой информацией, ее сначала нужно поместить во внутреннюю память. Например, прочитать номер телефона в записной книжке, а потом использовать эту информацию по назначению (набрать номер на аппарате).

Наша внутренняя память не всегда надежна. Человек нередко что-то забывает. Информация на внешних носителях хранится дольше и надежнее. Именно с помощью внешних носителей (перфокарты и перфоленты, магнитные ленты и магнитные диски, лазерные диски, флэш-память) люди передают свои знания из поколения в поколение.

Использование информации – это обязательный элемент формирования целенаправленной деятельности.



3. Измерение информации.

Любая наука рано или поздно приходит к необходимости как-то измерять то, что она изучает. Измерение информации – это одна из важнейших задач теоретической информатики.

Чтобы измерить информацию, нужно выбрать какую-то единицу измерения, эталон. В качестве такого эталона принимают информацию, полученную при выборе одного из двух вариантов.

Например, электрическая лампочка может находиться в двух состояниях: «горит» и «не горит». Тогда на вопрос «Горит ли сейчас лампочка» есть два возможных варианта ответа, которые можно обозначить цифрами 1 («горит») и 0 («не горит»). Поэтому ответ на этот вопрос может быть записан как 0 или 1.

Цифры 0 и 1 называют двоичными и с этим связано название единицы измерения количества информации – **бит**.

Английское слово **bit** – это сокращение от выражения **binary digit** – «*двоичная цифра*». Впервые слово бит в этом значении использовал американский инженер и математик Клод Шеннон в 1948 г.

Бит – это количество информации, соответствующее выбору одного из двух равновозможных вариантов.

Считать большие объемы информации в битах неудобно, хотя бы потому, что придется работать с очень большими числами (миллиардами, триллионами, и т.д.). Поэтому стоит ввести более крупные единицы.

Измерение количества информации тесно связано с устройством компьютерной памяти. Память строится из элементов, которые могут находиться в двух состояниях (0 или 1, включено или выключено). Поэтому информация о состоянии такого элемента равна 1 биту. Чтобы обращаться к ячейкам памяти, нужно каждой из них присвоить адрес (номер). Если каждый отдельный бит будет иметь свой адрес, адреса будут очень большие, и для их хранения потребуется много места. Кроме того, реальные данные состоят из нескольких битов, и каждый раз «собирать» число или символ из нескольких отдельных ячеек памяти неудобно. Поэтому группы соседних битов памяти объединяют в ячейки, каждая из которых имеет свой адрес и считывается (или записывается) как единое целое. Такие ячейки называются **байтами**.

Байт – это группа битов, имеющая собственный адрес в памяти.

Компьютер оперирует числами в **двоичной системе счисления**, поэтому в кратных единицах измерения количества информации используется коэффициент **2ⁿ**.

Так кратные байту единицы измерения количества информации вводятся следующим образом:

$$1 \text{ Кбайт} = 1024 \text{ байта} = 2^{10} \text{ байт}$$

$$1 \text{ Мбайт} = 1024 \text{ Кбайта} = 2^{20} \text{ байт}$$

$$1 \text{ Гбайт} = 1024 \text{ Мбайта} = 2^{30} \text{ байт}$$

$$1 \text{ Тбайт} = 1024 \text{ Гбайта} = 2^{40} \text{ байт}$$

$$1 \text{ Пбайт} = 1024 \text{ Тбайт} = 2^{50} \text{ байт}$$

В информатике используются различные подходы к измерению информации.

Содержательный подход к измерению информации

Информация – это знания человека. Сообщение *информативно* (содержит ненулевую информацию), если оно пополняет знания человека и *неинформативно* (количество информации равно нулю), если сведения старые, известные.

Можно различить две ситуации: «нет информации» - «есть информация» (т.е. количество информации равно нулю или не равно нулю).

Сообщение, уменьшающее неопределенность знаний в два раза, несет 1 бит информации.

Неопределенность знаний о некотором событии – это количество возможных результатов события.

Что такое «неопределенность знаний» рассмотрим на примере.

Допустим, вы бросаете монету, загадывая, что выпадет: орел или решка? Есть всего два варианта возможного результата бросания монеты. Причем не один из вариантов не имеет преимущества перед другим. В таком случае говорят, что они равновероятны. Так вот, в этом случае перед подбрасыванием монеты неопределенность знаний о результате равна двум.



После того, как вы бросили монету и посмотрели на нее, вы получили зрительное сообщение, что выпал, например, орел. Произошло одно из двух возможных событий. Неопределенность знаний уменьшилась в два раза: было два варианта, остался один. Значит, узнав результат бросания монеты, вы получили 1 бит информации.

Алфавитный подход к измерению информации

Объективным способом измерения информации является алфавитный подход. Только этот подход пригоден при использовании технических средств работы с информацией.

Алфавит – это все множество символов, используемых в некотором языке для представления информации.

Мощность алфавита – это число символов в алфавите.

Количество информации I , которое несет один символ в тексте и мощность алфавита N связаны формулой: $2^I = N$

Например, если считать, что появление символов в сообщении равновероятно, по формуле $N = 2^I$ можно рассчитать, какое количество информации несет каждый символ.

Так в русском алфавите, если не использовать букву ё, количество событий (букв) будет равно 32. Тогда: $32 = 2^I$, откуда $I = 5$ битов.

Каждый символ несет 5 битов информации. Количество информации в сообщении можно подсчитать, умножив количество информации, которое несет один символ. На количество символов.

Количество информации зависит от объема текста и от мощности алфавита.

Вероятностный подход к измерению информации

Все события происходят с различной вероятностью, но зависимость между вероятностью событий и количеством информации, полученной при совершении того или иного события можно выразить формулой, которую предложил К. Шеннон в 1948 году.

Формула 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

Для частного случая, когда события равновероятны ($p_i = 1/N$), величину количества информации I можно рассчитать по формуле:


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

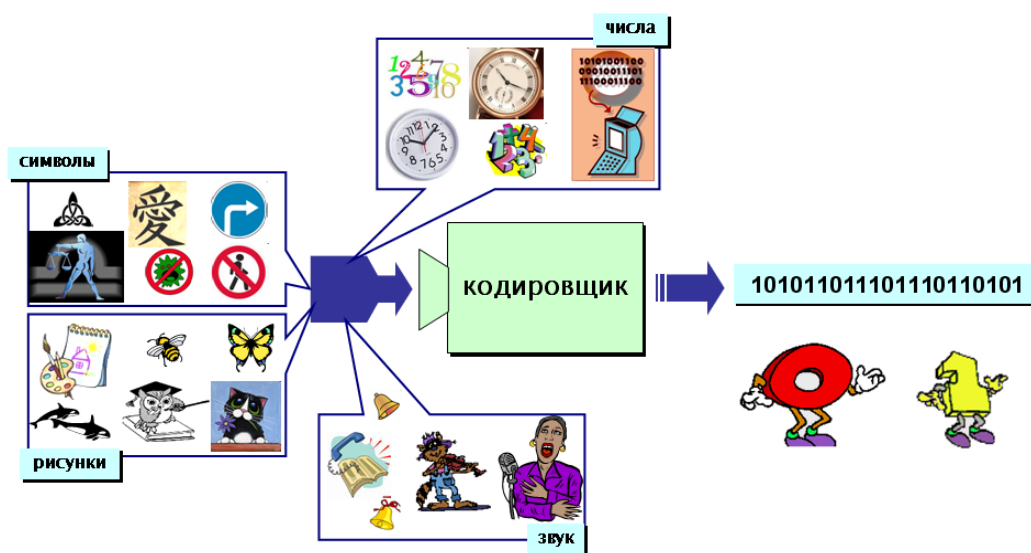
4. Кодирование информации

Для передачи и обработки информации ее всегда *кодируют*, то есть записывают в другой знаковой системе (на другом языке).

Язык – это система знаков, используемая для хранения, передачи и обработки информации.

Кодирование – это операция преобразование информации из одной формы представления (знаковой системы) в другую. Правило такого преобразования называется **кодом**.

Все виды информации (числовая, текстовая, графическая, звуковая) в компьютере кодируются на машинном языке, в виде логических последовательностей нулей и единиц. Такое кодирование называют **двоичным** (кодирование с помощью двух знаков).



Декодирование – это восстановление сообщения из последовательности кодов.

Кодирование числовой информации

Для записи информации о количестве объектов используются числа. Совокупность приемов наименования и записи чисел называется **счислением**.

Система счисления – это знаковая система, в которой числа записываются по определенным правилам с помощью символов некоторого алфавита, называемых цифрами.

Счисление представляет собой частный случай кодирования, где слово, записанное с использованием определенного алфавита и по определенным правилам, называется **кодом**. Применительно к счислению это **код числа**.

Различают *позиционные* и *непозиционные* системы счисления.

В **непозиционных системах счисления** каждое число обозначается соответствующей совокупностью символов. Характерным представителем *непозиционных* систем является *римская система счисления* со сложным способом записи чисел и громоздкими правилами выполнения арифметических операций. В этой системе счисления для записи чисел используются буквы латинского алфавита:

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

Значение цифры не зависит от ее положения в числе.

В **позиционных системах счисления** количественное значение цифры зависит от ее позиции в числе.

Наиболее распространенными в настоящее время позиционными системами счисления являются *десятичная, двоичная, восьмеричная* и *шестнадцатеричная*. Каждая позиционная система имеет определенный **алфавит цифр** и **основание**.

В **позиционных системах счисления** основание системы равно количеству цифр (знаков в ее алфавите) и определяет, во сколько раз различаются значения одинаковых цифр, стоящих в соседних позициях числа.

Позиционные системы счисления

Система счисления	Основание	Алфавит цифр
Десятичная	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Двоичная	2	0, 1
Восьмеричная	8	0, 1, 2, 3, 4, 5, 6, 7
Шестнадцатеричная	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A(10), B(11), C(12), D(13), E(14), F(15)

В позиционной системе счисления число в развернутой форме может быть представлено в следующем виде:

$$\text{✎ } X_{(q)} = x_{n-1}q^{n-1} + x_{n-2}q^{n-2} + \dots + x_1q^1 + x_0q^0 + x_{-1}q^{-1} + \dots + x_{-m}q^{-m}$$

где $X_{(q)}$ – запись числа в системе счисления с основанием q ;

q – основание системы счисления;

x_i – целые числа, меньше q ;

n – число разрядов (позиций) в целой части числа;

m – число разрядов в дробной части числа.

Таким образом, в любой системе счисления каждое число можно представить в развернутой записи, т.е. в виде суммы степеней основания, умноженных на коэффициент, соответствующий цифрам числа данной системы счисления.

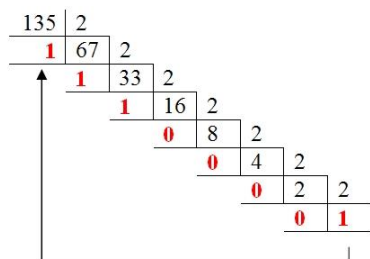
Например: в десятичной системе счисления число
 $1056,23_{(10)} = 1 \cdot 10^3 + 0 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0 + 2 \cdot 10^{-1} + 3 \cdot 10^{-2}$
 в двоичной системе счисления число
 $10010,11_{(2)} = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$
 в восьмеричной системе счисления число
 $145,541_{(8)} = 1 \cdot 8^2 + 4 \cdot 8^1 + 5 \cdot 8^0 + 5 \cdot 8^{-1} + 4 \cdot 8^{-2} + 1 \cdot 8^{-3}$
 в шестнадцатеричной системе счисления число
 $V3,AC_{(16)} = 11 \cdot 16^1 + 3 \cdot 16^0 + 10 \cdot 16^{-1} + 12 \cdot 16^{-2}$

Запятая отделяет целую часть числа от дробной и служит началом отсчета значений веса каждой позиции (разряда).

Перевод чисел из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную

Для перевода **целого числа** из десятичной системы счисления в другую необходимо последовательно делить это число на основание той системы, в которую оно переводится, пока частное не станет меньше делителя. Затем следует переписать, начиная с последнего частного, все остатки. Это и есть число в новой системе счисления.

Например, перевести число 135 из десятичной системы
в двоичную:



$$135_{10} = 10000111_2$$

в восьмеричную:

135	8	
7	16	8
	0	2

↑

$$135_{10} = 207_8$$

в шестнадцатеричную:

135	16
7	8

←

$$135_{10} = 87_{16}$$

Для перевода **правильной дроби** из десятичной системы счисления в другую необходимо последовательно умножать исходную дробь и дробные части получающихся произведений на основание той системы счисления, в которую она переводится. Затем целые части, полученные в результате умножения, следует записать в направлении от первой целой части. Перевод продолжается до тех пор, пока после очередного умножения дробная часть не окажется равной нулю или не будет достигнута заданная степень точности перевода.

Например, перевести дробь 0,45 из десятичной системы

в двоичную:	в восьмеричную:	в шестнадцатеричную:
0,45	0,45	0,45
× 2	× 8	× 16
0 90	3 60	7 20
× 2	× 8	× 16
1 80	4 80	3 20
× 2	× 8	× 16
1 60	6 40	3 20
× 2	× 8	× 16
1 20	3 20	3 20
↓	↓	↓
и т.д.	и т.д.	и т.д.

$$0,45_{10} = 0,0111_2$$

$$0,45_{10} = 0,3463_8$$

$$0,45_{10} = 0,7333_{16}$$

Для перевода смешанной дроби из десятичной системы счисления в другую необходимо перевести отдельно целую и дробную части по указанным выше правилам.

Перевод чисел в десятичную систему счисления

Для перевода числа из двоичной, восьмеричной или шестнадцатеричной системы счисления в десятичную необходимо это число представить в развернутой записи, т.е. в виде суммы степеней основания (2, 8 или 16), умноженных на коэффициенты, соответствующие цифрам переводимого числа, а затем рассчитать эту сумму.

Возьмем любое двоичное число, *например* $10000111,0111_2$. Запишем его в развернутой форме и произведем вычисления:

$$\begin{aligned} 10000111,0111_2 &= 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} = \\ &= 1 \cdot 128 + 0 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 + 0 \cdot 1/2 + 1 \cdot 1/4 + 1 \cdot 1/8 + 1 \cdot 1/16 = \\ &= 128 + 4 + 2 + 1 + 1/4 + 1/8 + 1/16 \approx 135,44_{10} \end{aligned}$$

Перевод числа из восьмеричной и шестнадцатеричной системы счисления в десятичную выполняется аналогично.

Перевод чисел между системами счисления, основания которых являются степенями числа 2 ($q=2^n$), может производиться по более простым алгоритмам. Такие алгоритмы могут применяться для перевода чисел между двоичной ($q=2^1$), восьмеричной ($q=2^3$) и шестнадцатеричной ($q=2^4$) системами счисления.

Перевод чисел из двоичной системы счисления в восьмеричную

Для записи двоичных чисел используются две цифры, то есть в каждом разряде числа возможны 2 варианта записи. Решаем показательное уравнение:

$$2 = 2^I. \text{ Так как } 2 = 2^1, \text{ то } I = 1 \text{ бит.}$$

Каждый разряд двоичного числа содержит 1 бит информации.

Для записи восьмеричных чисел используются восемь цифр, то есть в каждом разряде числа возможны 8 вариантов записи. Решаем показательное уравнение:

$$8 = 2^I. \text{ Так как } 8 = 2^3, \text{ то } I = 3 \text{ бита.}$$

Каждый разряд восьмеричного числа содержит 3 бита информации.

Таким образом, для перевода целого двоичного числа в восьмеричное его нужно разбить на группы по три цифры, справа налево, а затем преобразовать каждую группу в восьмеричную цифру. Если в последней, левой группе окажется меньше трех цифр, то необходимо ее дополнить слева нулями.

Для упрощения перевода используют таблицу преобразования двоичных **триад** (группа по три цифры) в восьмеричные цифры:

Двоичные триады	000	001	010	011	100	101	110	111
Восьмеричные цифры	0	1	2	3	4	5	6	7

Например, $10\ 000\ 111_2 = 010\ 000\ 111_2 = 207_8$

Для перевода дробного двоичного числа (правильной дроби) в восьмеричное необходимо разбить его на триады слева направо и, если в последней, правой группе окажется меньше трех цифр, дополнить ее справа нулями. Далее необходимо триады заменить на восьмеричные числа.

Например, $0,011\ 100\ 110\ 011_2 = 0,3463_8$

Перевод чисел из двоичной системы счисления в шестнадцатеричную

Для записи шестнадцатеричных чисел используются шестнадцать цифр, то есть в каждом разряде числа возможны 16 вариантов записи. Решаем показательное уравнение:

$$16 = 2^I. \text{ Так как } 16 = 2^4, \text{ то } I = 4 \text{ бита.}$$

Каждый разряд шестнадцатеричного числа содержит 4 бита информации.

Таким образом, для перевода целого двоичного числа в шестнадцатеричное его нужно разбить на группы по четыре цифры (**тетрады**), начиная справа, и, если в последней, левой группе окажется меньше четырех цифр, дополнить ее слева нулями. Для перевода дробного двоичного числа (правильной дроби) в шестнадцатеричное необходимо разбить его на тетрады слева направо и, если в последней, правой группе окажется меньше четырех цифр, то необходимо дополнить ее справа нулями.

Затем надо преобразовать каждую группу в шестнадцатеричную цифру, воспользовавшись для этого таблицей соответствия двоичных тетрад и шестнадцатеричных цифр.

Двоичные тетрады	0000	0001	0010	0011	0100	0101	0110	0111
Шестнадцатеричные цифры	0	1	2	3	4	5	6	7
Двоичные тетрады	1000	1001	1010	1011	1100	1101	1110	1111
Шестнадцатеричные цифры	8	9	A	B	C	D	E	F

Например, $1000\ 0111_2 = 87_{16}$

$$0,0111\ 0011\ 0011\ 0011_2 = 0,7333_{16}$$

Кодирование текстовой информации

Для кодирования букв и ряда специальных знаков достаточно использовать 256 различных символов. По формуле, связывающей количество сообщений N и количество информации I , можно вычислить, какое количество информации необходимо, чтобы закодировать каждый знак:

$$N = 2^I \Rightarrow 256 = 2^I \Rightarrow 2^8 = 2^I \Rightarrow I = 8 \text{ битов} = 1 \text{ байт}$$

Традиционно для кодирования одного символа используется количество информации, равное 1 байту, то есть $I = 1 \text{ байт} = 8 \text{ битов}$.

Кодирование заключается в том, что каждому символу ставится в соответствие уникальный десятичный код от 0 до 255 или соответствующий ему двоичный код от 00000000 до 11111111. Таким образом, человек различает символы по их начертанию, а компьютер – по их коду.

При вводе в компьютер текстовой информации происходит ее двоичное кодирование, изображение символа преобразуется в его двоичный код. Пользователь нажимает на клавиатуре клавишу с символом, и в компьютер поступает определенная последовательность из восьми электрических импульсов (двоичный код символа). Код символа хранится в оперативной памяти компьютера, где занимает одну ячейку – 1 байт.

Присвоение символу конкретного кода – это вопрос соглашения, которое фиксируется в *кодовой таблице*.

КОДОВЫЕ ТАБЛИЦЫ

- КОИ–7, КОИ–8 – кодирование русских букв и символов (семи-, восьми-битное кодирование)

–		Г	Г	Л	Л	Т	Т	Т	Т	Т	■	■	■	■	■
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
▒	▒	▒	Г	■	●	√	≈	≤	≥	nbsp	↓	◦	²	•	÷
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
=		ƒ	ё	п	Г	г	п	Г	Е	Ц	Ц	Г	Л	Г	Г
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
Г	Г	Г	ё	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	Г	©
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	Ъ
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

1) #154 – неразрывный пробел

Рис.2. Кодировка КОИ8-Р

- **ASCII** – American Standard Code for Information Interchange (американский стандарт кодов для обмена информацией) – это восьмиразрядная кодовая таблица, в ней закодировано 256 символов (127 – стандартные коды символов английского языка, спецсимволы, цифры, а коды от 128 до 255 – национальный стандарт, алфавит языка, символы псевдографики, научные символы, коды от 0 до 32 отведены не символам, а функциональным клавишам).

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

1) #32 – пробел

Рис. 3. Международная кодировка ASCII

- **Unicode** – стандарт, согласно которому для представления каждого символа используется 2 байта (можно кодировать математические символы, русские, английские, греческие, и даже китайские). С его помощью можно закодировать не 256, а 65536 различных символов. Полная спецификация стандарта Unicode включает в себя все существующие, вымершие и искусственно созданные алфавиты мира, а также множество математических, музыкальных, химических и прочих символов.
- **CP1251** – наиболее распространенной в настоящее время является кодировка Microsoft Windows, (CP – означает «Code Page – кодовая страница»).

Á	à	,	è	„	…	ř	ř	€	‰	É	<	й	Й	ó	ú
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
á	‘	’	“	”	•	–	—	è	™	é	>	ò	й	ó	ú
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
nbsp	ÿ	Ы	Э	х	ы	ı	ş	Ë	©	Ю	<<	¬	shy	®	Я
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
°	±	Ы	Э	’	µ	¶	•	ë	№	ю	>>	э	ю	я	я
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

1) #160 – неразрывный пробел

2) #173 – мягкий перенос

Рис. 4. Кодировка CP1251

- CP866 – кодировка под MS DOS

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
▒	▓	█		†	‡	§	¶	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
Ё	ё	Є	є	Ї	ї	Ÿ	ÿ	•	•	•	√	№	¤	■	nbsp
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

1) #255 – неразрывный пробел

Рис. 5. Кодировка CP866

- Mac – кодировка в ПК фирмы Apple, работающих под управлением операционной системы Mac OS.

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
ı	°	Ы	£	§	•	¶	Ы	©	©	™	Á	á	ë	à	è
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
Ё	±	≤	≥	э	μ	г	Э	Ю	ю	Я	я	É	é	Й	ò
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
э	ю	¬	√	f	≈	Δ	«	»	...	nbsp	Ó	ó	Й	й	я
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
–	—	“	”	‘	’	÷	„	ÿ	Ы	У	У	№	Ё	ё	я
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

1) #202 – неразрывный пробел.

Рис. 6. Кодировка Mac

- ISO 8859– 5 – Международная организация по стандартизации (International Standards Organization, ISO) утвердила в качестве стандарта для русского языка еще одну кодировку.

І	І	І	І	І	І	І	І	І	І	І	І	І	І	І	І
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
І	І	І	І	І	І	І	І	І	І	І	І	І	І	І	І
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
nbsp	Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	shy	Ў	Ц
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
№	ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ќ	ќ	ѕ	ў	ц
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

- 1) Коды 128 – 159 не используются;
- 2) #160 – неразрывный пробел,
- 3) #173 – мягкий перенос.

Рис. 7. Кодировка ISO 8859-5

Кодирование графической информации

Графическая информация на экране монитора представляется в виде *растрового изображения*, которое формируется из определенного количества точек – **пикселей** (англ. pixel образовано от словосочетания picture element, что означает **элемент изображения**).

Качество изображения определяется **разрешающей способностью монитора**, то есть количеством точек, из которых оно складывается. Чем больше разрешающая способность, то есть чем больше количество строк раstra и точек в строке, тем выше качество изображения. В современных персональных компьютерах обычно используются три основные разрешающие способности экрана: 800 × 600, 1024 × 768 и 1280 × 1024 точки.

Для хранения **черно-белого** изображения используется **1 бит**.

Цветные изображения формируются в соответствии с двоичным кодом цвета каждой точки, хранящимся в видеопамяти. Цветные изображения имеют различную **глубину цвета**, которая задается количеством битов, используемым для кодирования цвета точки. Наиболее распространенными значениями глубины цвета являются 8, 16, 24 или 32 бита.

Качество двоичного кодирования изображения определяется **разрешающей способностью экрана** и **глубиной цвета**.

Каждый цвет можно рассматривать как возможное состояние точки, тогда количество цветов, отражаемых на экране монитора, может быть вычислено по формуле:

$$N = 2^I,$$

где I – глубина цвета.

Глубина цвета и количество отображаемых цветов

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

Цветное изображение на экране формируется за счет смешивания трех базовых цветов: **красного**, **зеленого** и **синего**. Такая цветовая модель называется **RGB-моделью** по первым буквам английских названий цветов (**Red, Green, Blue**).

Для получения богатой палитры цветов базовым цветам могут быть заданы различные интенсивности. *Например*, при глубине цвета в 24 бита на каждый из цветов выделяется 8 бит, то есть для каждого из цветов возможны $N = 2^8 = 256$ уровней интенсивности, заданные двоичными кодами (от минимальной – 00000000 до максимальной – 11111111).

Формирование цветов при глубине цвета 24 бита

Название цвета	Интенсивность		
	красный	зеленый	синий
Черный	00000000	00000000	00000000
Красный	11111111	00000000	00000000
Зеленый	00000000	11111111	00000000
Синий	00000000	00000000	11111111
Голубой	00000000	11111111	11111111
Желтый	11111111	11111111	00000000
Белый	11111111	11111111	11111111

Графический режим вывода изображения на экран монитора определяется величиной разрешающей способности и глубиной цвета. Для того чтобы на экране монитора формировалось изображение, информация о каждой его точке (код цвета точки) должна храниться в видеопамяти компьютера.

Кодирование звуковой информации

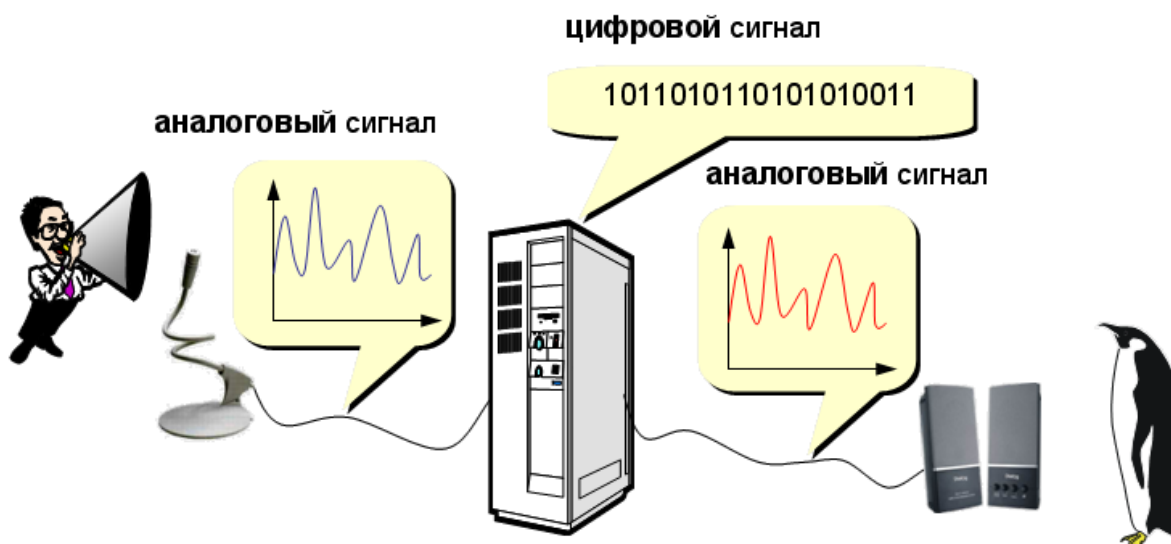
Звук представляет собой звуковую волну с непрерывно меняющейся амплитудой и частотой. Чем больше амплитуда сигнала, тем он громче для человека, чем больше частота сигнала, тем выше тон. Для того чтобы компьютер мог обрабатывать звук, непрерывный звуковой сигнал должен быть превращен в последовательность электрических импульсов (двоичных нулей и единиц).

В процессе кодирования непрерывного звукового сигнала производится его **временная дискретизация**. Непрерывная звуковая волна разбивается на отдельные маленькие временные участки, причем для каждого такого участка устанавливается определенная величина амплитуды. Таким образом, непрерывная зависимость амплитуды сигнала от времени заменяется на дискретную последовательность уровней громкости. Уровни громкости звука можно рассматривать как набор возможных состояний, соответственно, чем больше количество уровней громкости будет выделено в процессе кодирования, тем больше количество информации будет нести значение каждого уровня и тем более качественным будет звучание.

Современные звуковые карты обеспечивают 16-битную глубину кодирования звука.

При двоичном кодировании непрерывного звукового сигнала он заменяется последовательностью дискретных уровней сигнала. Качество кодирования зависит от количества измерений уровня сигнала в единицу времени, то есть **частоты дискретизации**. Чем больше количество измерений производится за 1 секунду (чем больше частота дискретизации), тем точнее процедура двоичного кодирования.

Качество двоичного кодирования звука определяется **глубиной кодирования** и **частотой дискретизации**.



Контрольные вопросы

1. Расскажите об информатике как об отрасли, как о науке, как о прикладной дисциплине. Опишите цели и задачи.
2. Дайте определение термину «информация». Какие подходы в определении этого термина вам известны.
3. Чем информация отличается от данных? Как связаны между собой эти два понятия?
4. По каким признакам можно классифицировать информацию?
5. Что такое «адекватность информации»? Какие виды адекватности вы знаете?
6. Каким образом измеряют информацию? Какие меры информации вам известны?
7. Как связаны между собой количество информации и мера неопределенности состояния системы?
8. Дайте определения информационным процессам.
9. Нарисуйте схему передачи информации и поясните назначение ее элементов.
10. Что такое «кодирование» и «декодирование»?

Лекция 2.

ЛОГИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

1. *Формы мышления.*
2. *Логические операции над высказываниями.*
3. *Логические выражения и таблицы истинности.*

Основы формальной логики заложил Аристотель, который впервые отделил логические формы мышления (речи) от его содержания.

Логика – это наука о формах и способах мышления.

1. Формы мышления

Мышление всегда осуществляется в каких-то формах.

Основными формами мышления являются **понятие**, **высказывание** и **умозаключение**.

Понятие – это форма мышления, фиксирующая основные, существенные признаки объекта.

Понятие имеет две стороны: **содержание** и **объем**.

Содержание понятия составляет совокупность существенных признаков объекта.

Например, содержание понятия «персональный компьютер» можно раскрыть следующим образом: «Персональный компьютер – это универсальное электронное устройство для автоматической обработки информации, предназначенное для одного пользователя».

Объем понятия определяется совокупностью предметов, на которую оно распространяется.

Объем понятия «персональный компьютер» выражает всю совокупность существующих в настоящее время в мире персональных компьютеров.

Свое понимание окружающего мира человек формулирует в форме высказываний.

Высказывание – это форма мышления, в которой что-либо утверждается или отрицается о свойствах реальных предметов и отношениях между ними. Высказывание может быть либо *ИСТИННО*, либо *ЛОЖНО*.

Умозаключение – это форма мышления, с помощью которой из одного или нескольких суждений может быть получено новое суждение (заключение).

2. Логические операции над высказываниями

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

A, B – логические переменные

0 – *ложь* } значения логических переменных
 1 – *истина* }

Существуют разные варианты обозначения истинности и ложности переменных:

Истина	И	1	True	T
Ложь	Л	0	False	F

В алгебре логики над высказываниями можно производить определенные логические операции, в результате которых получаются новые, составные высказывания.

Логическое отрицание (инверсия)

Присоединение частицы «не» к высказыванию называется операцией *логического отрицания* или *инверсией*.

Логическое отрицание (инверсия) делает истинное высказывание ложным и, наоборот, ложное – истинным.

Операцию логического отрицания (инверсию) над логическим высказыванием A в алгебре логики принято обозначать \bar{A} .

$\neg A$ или \bar{A} (не A)

Таблица истинности логического отрицания

A	\bar{A}
1	0
0	1

Логическое умножение (конъюнкция)

Объединение двух (или нескольких) высказываний в одно с помощью союза «и» называется операцией **ЛОГИЧЕСКОГО УМНОЖЕНИЯ** или **КОНЪЮНКЦИЕЙ**.

Составное высказывание, образованное в результате **операции логического умножения (конъюнкции)**, истинно тогда и только тогда, когда истинны все входящие в него простые высказывания.

Операцию логического умножения принято обозначать \wedge или $\&$.

$A \wedge B$ или $A \& B$ (A и B)

Таблица истинности логического умножения

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Логическое сложение (дизъюнкция)

Объединение двух (или нескольких) высказываний с помощью союза «или» называется операцией **ЛОГИЧЕСКОГО СЛОЖЕНИЯ** или **ДИЗЪЮНКЦИЕЙ**.

Составное высказывание, образованное в результате **операции логического сложения (дизъюнкции)**, истинно тогда, когда истинно хотя бы одно из входящих в него простых высказываний.

Операцию логического сложения принято обозначать \vee или $+$.

$A \vee B$ или $A + B$ (A или B)

Таблица истинности логического сложения

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

Логическое следование (импликация) образуется соединением двух высказываний в одно с помощью оборота речи «если ..., то ...».

Операцию импликация принято обозначать \rightarrow

$A \rightarrow B$ (если A, то B; из A, следует B; A влечет за собой B)

Таблица истинности логического следования

A	B	A \rightarrow B
0	0	1
0	1	1
1	0	0
1	1	1

Логическое равенство (эквивалентность) образуется соединением двух высказываний в одно с помощью оборота речи «... тогда и только тогда, когда ...».

Операцию эквивалентность принято обозначать \leftrightarrow или \sim

$A \leftrightarrow B$ или $A \sim B$ (A тогда и только тогда, когда B)

Таблица истинности логического равенства

A	B	A \leftrightarrow B
0	0	1
0	1	0
1	0	0
1	1	1

Неравенство (неэквивалентность) – A неэквивалентно B (исключающее ИЛИ – логическая операция, результат выполнения операции является истинным только при условии, если является истинным в точности один из аргументов).

Операцию неэквивалентность («исключающее или») принято обозначать \oplus

$$A \oplus B \text{ (} A \text{ неэквивалентно } B \text{)}$$

Таблица истинности логического неравенства

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

При вычислении логического выражения операции выполняются в следующем порядке:

- **инверсия** (отрицание)
- **конъюнкция** (логическое умножение)
- **дизъюнкция** (логическое сложение)
- **импликация** (логическое следование)
- **эквивалентность** (логическое равенство)
- **исключающее или** (отрицание эквивалентности)

Как и в обычной алгебре, для изменения порядка операций используются скобки.

Основные законы алгебры логики

Название закона	Формулировка
Переместительный закон	$A \vee B = B \vee A$ $A \wedge B = B \wedge A$
Сочетательный закон	$(A \vee B) \vee C = A \vee (B \vee C)$ $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
Распределительный закон	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
Закон непротиворечия Этот закон выражает тот факт, что высказывание не может быть одновременно истинным и ложным	$A \wedge \neg A = 0$

Закон исключенного третьего Этот закон означает, что либо высказывание, либо его отрицание должно быть истинным	$A \vee \neg A = 1$
Закон двойного отрицания	$\neg(\neg A) = A$
Законы де Моргана	$\neg(A \vee B) = \neg A \wedge \neg B$ $\neg(A \wedge B) = \neg A \vee \neg B$

Основные свойства конъюнкции, дизъюнкции и импликации

$$A \wedge A = A$$

$$A \wedge 0 = 0$$

$$A \vee A = A$$

$$A \vee 1 = 1$$

$$A \vee 0 = A$$

$$A \wedge 1 = A$$

$$0 \rightarrow A = 1$$

3. Логические выражения и таблицы истинности

Логические выражения. Каждое составное высказывание можно выразить в виде формулы (логического выражения), в которую входят *логические переменные*, обозначающие высказывания, и *знаки логических операций*, обозначающие логические функции.

Запишем в форме логического выражения составное высказывание

$$2 \cdot 2 = 5 \text{ или } 2 \cdot 2 = 4 \text{ и } 2 \cdot 2 \neq 5 \text{ или } 2 \cdot 2 \neq 4$$

Проанализируем составное высказывание. Оно содержит два простых высказывания:

$$A = 2 \cdot 2 = 5 \text{ — ложно — } (0)$$

$$B = 2 \cdot 2 = 4 \text{ — истинно — } (1)$$

Тогда составное высказывание можно записать

$$(A \text{ или } B) \text{ и } (\bar{A} \text{ или } \bar{B})$$

Запишем высказывание в форме логического выражения

$$(A \vee B) \wedge (\bar{A} \vee \bar{B})$$

При выполнении логических операций определен следующий порядок их выполнения: инверсия, конъюнкция, дизъюнкция. Для изменения указанного порядка могут использоваться скобки.

Подставим в логическое выражение значения логических переменных и, используя таблицы истинности, получим значение логической функции:

$$F = (A \vee B) \wedge (\bar{A} \vee \bar{B}) = (0 \vee 1) \wedge (1 \vee 0) = 1 \wedge 1 = 1$$

Таблицы истинности. Для каждого составного высказывания (логического выражения) можно построить таблицу истинности, которая определяет его истинность или ложность при всех возможных комбинациях исходных значений простых высказываний (логических переменных).

При построении таблиц истинности целесообразно руководствоваться определенной последовательностью действий.

Во-первых, необходимо определить количество строк в таблице истинности. Оно равно количеству возможных комбинаций значений логических переменных, входящих в логическое выражение. Если количество логических переменных равно n , то:

$$\text{количество строк} = 2^n$$

В нашем случае логическая функция F имеет 2 переменные и, следовательно, количество строк в таблице истинности должно быть равно 4.

Во-вторых, необходимо определить количество столбцов в таблице истинности, которое равно количеству логических переменных плюс количество логических операций.

В нашем случае количество переменных равно двум, а количество логических операций – пяти, то есть количество столбцов таблицы истинности равно семи.

В-третьих, необходимо построить таблицу истинности с указанным количеством строк и столбцов, обозначить столбцы и внести в таблицу возможные наборы значений исходных логических переменных.

В-четвертых, необходимо заполнить таблицу истинности по столбцам, выполняя базовые логические операции в необходимой последовательности.

Таблица истинности логической функции $F = (A \vee B) \wedge (\bar{A} \vee \bar{B})$

A	B	$A \vee B$	\bar{A}	\bar{B}	$\bar{A} \vee \bar{B}$	$(A \vee B) \wedge (\bar{A} \vee \bar{B})$
0	0	0	1	1	1	0
0	1	1	1	0	1	1
1	0	1	0	1	1	1
1	1	1	0	0	0	0

Равносильные логические выражения. Логические выражения, у которых последние столбцы таблиц истинности совпадают, называются **равносильными**. Для обозначения равносильных логических выражений используется знак $=$.

Доказать, что логические выражения $\bar{A} \wedge \bar{B}$ и $\overline{A \vee B}$ равносильны.

Построим сначала таблицу истинности логического выражения $\bar{A} \wedge \bar{B}$

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

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

Построим таблицу истинности логического выражения $\overline{A \vee B}$

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

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

Значения в последних столбцах таблиц истинности совпадают, следовательно, логические выражения равносильны $\bar{A} \wedge \bar{B} = \overline{A \vee B}$

Контрольные вопросы

1. Что изучает логика?
2. Какие существуют основные формы мышления?
3. В чем состоит разница между содержанием и объемом понятия?
4. Может ли быть высказывание выражено в форме вопросительного предложения?
5. Как определяется истинность или ложность простого высказывания? Составного высказывания?
6. Какие законы логики вы знаете?
7. Что содержат таблицы истинности и каков порядок их построения?
8. Какие логические выражения называются равносильными?
9. Какое количество логических функций двух аргументов существует и почему?
10. Какие логические функции двух аргументов имеют свои названия?

Лекция 3.

ТЕХНИЧЕСКИЕ СРЕДСТВА ОБРАБОТКИ ИНФОРМАЦИИ

1. Классическая архитектура ЭВМ.
2. Магистрально-модульный принцип построения персонального компьютера.
3. Базовая аппаратная конфигурация персонального компьютера.
4. Периферийные устройства персонального компьютера.

1. Классическая архитектура ЭВМ

С середины 60-х годов существенно изменился подход к созданию вычислительных машин. Вместо независимой разработки аппаратуры и некоторых средств математического обеспечения стала проектироваться система, состоящая из совокупности *аппаратных (hardware)* и *программных (software)* средств. При этом на первый план выдвинулась концепция их взаимодействия. Так возникло принципиально новое понятие – архитектура ЭВМ.

Архитектура ЭВМ – концептуальная структура вычислительной машины, определяющая проведение обработки информации и включающая методы преобразования информации в данные и принципы взаимодействия технических средств и программного обеспечения.

Архитектуру вычислительного средства следует отличать от его структуры. **Структура вычислительного средства** определяет его конкретный состав на некотором уровне детализации (устройства, блоки узлы и т.д.) и описывает связи внутри средства во всей их полноте. Архитектура же определяет правила взаимодействия составных частей вычислительного средства, описание которых выполняется в той мере, в какой это необходимо для формирования правил их взаимодействия. Она регламентирует не все связи, а наиболее важные, которые должны быть известны для более грамотного использования данного средства.

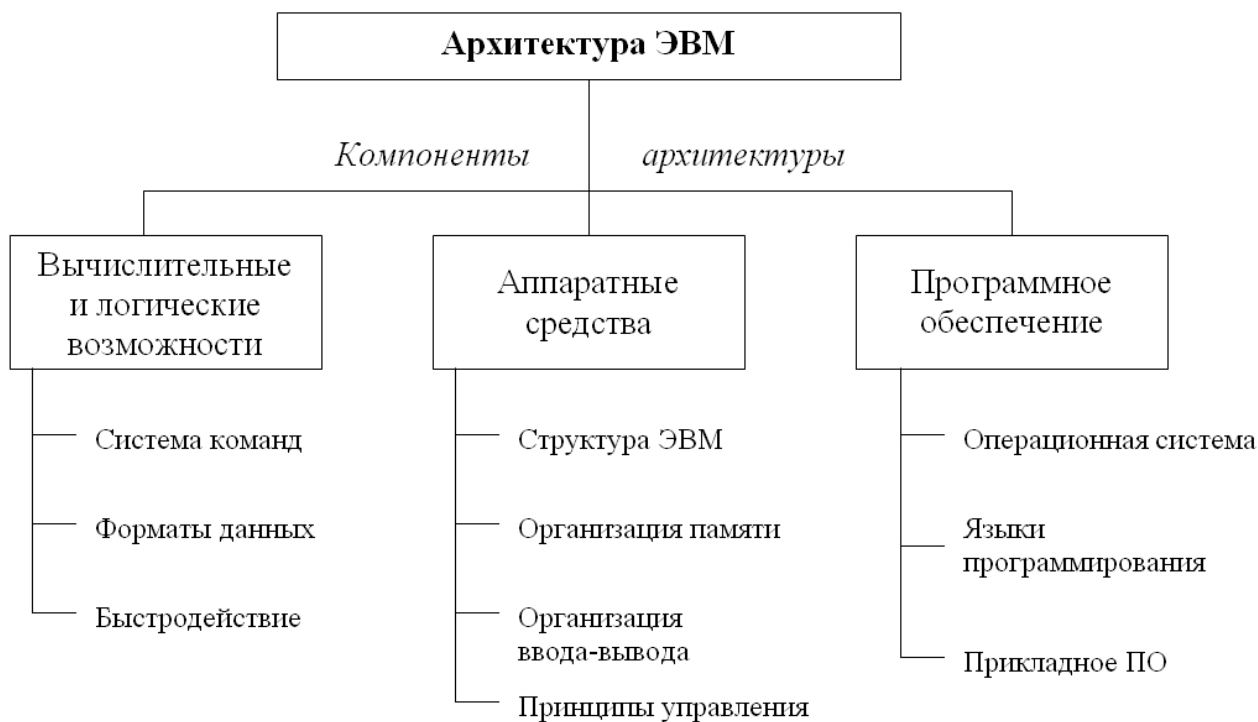


Рис. 1. Основные компоненты архитектуры ЭВМ

Архитектура ЭВМ действительно отражает круг проблем, относящихся к общему проектированию и построению вычислительных машин и их программного обеспечения.

В настоящее время наибольшее распространение в ЭВМ получили 2 типа архитектуры: **принстонская** (фон Неймана) и **гарвардская**. Обе они выделяют 2 основных узла ЭВМ: *центральный процессор* и *память* компьютера. Различие заключается в структуре памяти: в принстонской архитектуре программы и данные хранятся в одном массиве памяти и передаются в процессор по одному каналу, тогда как гарвардская архитектура предусматривает отдельные хранилища и потоки передачи для команд и данных.

Историческая справка. В 30-х годах правительство США поручило Гарвардскому и Принстонскому университетам разработать архитектуру компьютера для военно-морской артиллерии. Победила разработка Принстонского университета (более известная как архитектура фон Неймана, названная так по имени разработчика, первым предоставившего отчет об архитектуре), так как она была проще в реализации.

Основы учения об архитектуре ЭВМ заложил выдающийся американский математик **Джон фон Нейман**. Он предложил **основные принципы по-**

строения ЭВМ: переход к двоичной системе счисления для представления информации и принцип хранимой программы.

Программу вычислений предлагалось помещать в запоминающем устройстве ЭВМ, что обеспечивало бы автоматический режим выполнения команд и, как следствие, увеличение быстродействия ЭВМ. (*Ранее все вычислительные машины хранили обрабатываемые числа в десятичном виде, а программы задавались путем установки переключателей на специальной коммутационной панели*). Нейман первым догадался, что программа может также храниться в виде набора нулей и единиц, причём в той же памяти, что и обрабатываемые ею числа.

Основные принципы построения ЭВМ:

1. Любую ЭВМ образуют три основных компонента: процессор, память и устройства ввода-вывода.
2. Информация, с которой работает ЭВМ, делится на два типа:
 - набор команд по обработке (программы);
 - данные подлежащие обработке.
3. И команды, и данные вводятся в память (ОЗУ) – ***принцип хранимой программы***.
4. Руководит обработкой процессор, устройство управления (УУ) которого выбирает команды из ОЗУ и организует их выполнение, а арифметико-логическое устройство (АЛУ) проводит арифметические и логические операции над данными.
5. С процессором и ОЗУ связаны устройства ввода-вывода (УВВ).

Фон Нейман не только выдвинул основополагающие принципы логического устройства ЭВМ, но и предложил структуру, которая воспроизводилась в течение первых двух поколений ЭВМ. Разработанные фон Нейманом основы архитектуры вычислительных устройств оказались настолько фундаментальными, что получили в литературе название «фон Неймановской архитектуры».



Рис. 2. Архитектура ЭВМ, построенная на принципах фон Неймана

Арифметико-логическое устройство (АЛУ) и устройство управления (УУ) объединены в общее устройство, называемое центральным процессором.

Процессор, или **микروпроцессор**, является основным устройством ЭВМ. Он предназначен для выполнения вычислений по хранящейся в запоминающем устройстве программе и обеспечения общего управления ЭВМ. Быстродействие ЭВМ в значительной мере определяется скоростью работы процессора.

Вычислительный процесс должен быть предварительно представлен для ЭВМ в виде программы – последовательности инструкций (команд), записанных в порядке выполнения. В процессе выполнения программы ЭВМ выбирает очередную команду, расшифровывает ее, определяет, какие действия и над какими операндами следует выполнить. Эту функцию осуществляет устройство управления (УУ). Оно же помещает выбранные из запоминающего устройства (ЗУ) операнды в арифметико-логическое устройство (АЛУ), где они и обрабатываются. Само АЛУ работает под управлением УУ.

Обрабатываемые данные и выполняемая программа должны находиться в **запоминающем устройстве – памяти ЭВМ**, куда они вводятся через устройство ввода. Емкость памяти измеряется в величинах, кратных байту. Память представляет собой сложную структуру, построенную по иерархическому принципу, и включает в себя запоминающие устройства различных типов. Функционально она делится на две части: внутреннюю и внешнюю.

Внутренняя память – это запоминающее устройство, напрямую связанное с процессором и предназначенное для хранения выполняемых программ и данных, непосредственно участвующих в вычислениях.

Внутренняя память, в свою очередь, делится на *оперативную (ОЗУ)* и *постоянную (ПЗУ) память*.

В **оперативном запоминающем устройстве (ОЗУ)** в двоичном виде запоминается обрабатываемая информация, программа ее обработки, промежуточные данные и результаты работы. ОЗУ обеспечивает режимы записи, считывания и хранения информации, причем в любой момент времени возможен доступ к любой произвольно выбранной ячейке памяти. Это отражено в англоязычном названии ОЗУ – **RAM** (Random Access Memory – память произвольного доступа). Доступ к этой информации в ОЗУ осуществляется очень быстро. Эта память составлена из сложных электронных микросхем и расположена внутри корпуса компьютера. Часть оперативной памяти отводится для хранения изображений, получаемых на экране монитора, и называется **видеопамять**. Чем больше видеопамять, тем более сложные и качественные картинки может выводить компьютер. Высокоскоростная **кэш-память** служит для увеличения скорости выполнения операций компьютером и используется при обмене данными между микропроцессором и RAM. Кэш-память является промежуточным запоминающим устройством (буфером). Существует два вида кэш-памяти: внутренняя, размещаемая внутри процессора и внешняя, размещаемая на материнской плате. Оперативная память является энергозависимой, т.е. при выключении питания ЭВМ содержимое оперативной памяти в большинстве случаев теряется.

Постоянная память (ПЗУ) обеспечивает *хранение и выдачу* информации. В отличие от содержимого оперативной памяти, содержимое постоянной заполняется при изготовлении ЭВМ и не может быть изменено в обычных условиях эксплуатации. В постоянной памяти хранятся часто используемые

(универсальные) программы и данные, к примеру, некоторые программы операционной системы, программы тестирования оборудования ЭВМ и др. При выключении питания содержимое постоянной памяти сохраняется.

Внешняя память (ВЗУ) предназначена для размещения больших объемов информации и обмена ею с оперативной памятью. В качестве ВЗУ используют накопители на магнитных лентах, на магнитных и оптических дисках, флэш-дисках.

Внешняя память может быть с **произвольным доступом** (накопители на магнитных и оптических дисках) и **последовательным доступом** (накопители на магнитных лентах, например, стримеры). *Устройства памяти с произвольным доступом* позволяют получить доступ к произвольному блоку данных примерно за одно и то же время доступа. *Устройства памяти с последовательным доступом* позволяют осуществлять доступ к данным последовательно, т.е. для того, чтобы считать нужный блок памяти, необходимо считать все предшествующие блоки.

Устройства с произвольным доступом обладают большим быстродействием, поэтому они являются основными внешними запоминающими устройствами, постоянно используемыми в процессе функционирования ЭВМ. Устройства последовательного доступа используются в основном для резервирования информации.

Устройства ввода-вывода служат соответственно для ввода информации в ЭВМ и вывода из нее, а также для обеспечения общения пользователя с машиной. Процессы ввода-вывода протекают с использованием внутренней памяти ЭВМ. Иногда устройства ввода-вывода называют *периферийными* или *внешними устройствами* ЭВМ. К ним относятся, в частности, мониторы, клавиатура, манипуляторы, принтеры, графопостроители, сканеры и др.

Пульт управления служит для выполнения системных операций в ходе управления вычислительным процессом. Пульт управления конструктивно часто выполняется вместе с центральным процессором.

2. Магистрально-модульный принцип построения персонального компьютера

В основу архитектуры современных персональных компьютеров положен *магистрально-модульный принцип*. Модульная организация компьютера

опирается на магистральный (шинный) принцип обмена информацией между устройствами.

Магистраль (*системная шина*) включает в себя три многозарядные шины: *шину данных*, *шину адреса* и *шину управления*, которые представляют собой многопроводные линии.



Рис. 3. Магистрально-модульный принцип построения персонального компьютера

К магистральной подключаются процессор и оперативная память, а также периферийные устройства ввода, вывода и хранения информации, которые обмениваются информацией на машинном языке (последовательностями нулей и единиц в форме электрических импульсов).

Шина данных. По этой шине данные передаются между различными устройствами. *Например*, считанные из оперативной памяти данные могут быть переданы процессору для обработки, а затем полученные данные могут быть отправлены обратно в оперативную память для хранения. Таким образом, данные по шине данных могут передаваться от устройства к устройству в любом направлении. *Разрядность шины данных* определяется *разрядностью процессора*, то есть количеством двоичных разрядов, которые могут обрабатываться или передаваться процессором за один такт. Разрядность процессоров постоянно увеличивается по мере развития компьютерной техники.

Шина адреса. Выбор устройства или ячейки памяти, куда пересылаются

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

Разрядность шины адреса определяет объем адресуемой памяти (адресное пространство), то есть количество однобайтовых ячеек оперативной памяти, которые могут иметь уникальные адреса. Количество адресуемых ячеек памяти можно рассчитать по формуле:

$$N = 2^I,$$

где I – разрядность шины адреса.

Шина управления. По шине управления передаются сигналы, определяющие характер обмена информацией по магистрали. Сигналы управления показывают, какую операцию – считывание или запись информации из памяти – нужно производить, синхронизируют обмен информацией между устройствами и так далее.

В современных ЭВМ реализован **принцип открытой архитектуры**, позволяющий пользователю самому комплектовать нужную ему конфигурацию компьютера и производить при необходимости ее модернизацию. Принцип открытой архитектуры позволяет менять состав устройств ЭВМ. К информационной магистрали могут подключаться дополнительные периферийные устройства, одни модели устройств могут заменяться на другие.

Внешние устройства к шинам подключаются посредством **интерфейса**. Под **интерфейсом** понимают совокупность различных характеристик какого-либо периферийного устройства персонального компьютера, определяющих организацию обмена информацией между ним и центральным процессором.

Аппаратное подключение периферийного устройства к магистрали на физическом уровне осуществляется через специальный блок – **контроллер** (другие названия – адаптер, плата, карта). Для установки контроллеров на материнской плате имеются специальные разъемы – **слоты**.

Управление работой периферийного устройства производится через программу – **драйвер**, которая является компонентой операционной системы.

Связь компьютера с внешними устройствами осуществляется через **порты** – специальные разъемы на задней панели компьютера. Различают *последовательные* и *параллельные* порты.

Последовательный порт (*например, СОМ – порт*) – двунаправленный последовательный интерфейс, предназначенный для обмена битовой информацией. Последовательным порт называется потому, что информация через него передается по одному биту, бит за битом (в отличие от параллельного порта). Последовательные порты служат для подключения манипуляторов, модема, а также он используется для соединения с источниками бесперебойного питания, для связи с аппаратными средствами разработки встраиваемых вычислительных систем.

Параллельный порт (*например, LPT – порты*) служит для подключения принтеров, сканеров и передает информацию по нескольким линиям – несколько битов передаются одновременно (параллельно).

В последнее время широкое распространение получили *последовательные универсальные порты* (USB), предназначенные для подключения периферийных устройств.

Процессор и оперативная память

Процессор. Процессор аппаратно реализуется на большой интегральной схеме (БИС). Большая интегральная схема представляет собой маленькую плоскую полупроводниковую пластину размером примерно 20×20 мм, заключенную в плоский корпус с рядами металлических контактов. БИС является «большой» по количеству элементов.

Важнейшей характеристикой, определяющей быстродействие процессора, является **тактовая частота**, то есть количество тактов в секунду. **Такт** – это промежуток времени между началами подачи двух последовательных импульсов специальной микросхемой – генератором тактовой частоты, синхронизирующим работу узлов компьютера. На выполнение процессором каждой базовой операции отводится определенное количество тактов. Чем больше тактовая частота, тем больше операций в секунду выполняет процессор. Тактовая частота измеряется в мегагерцах (МГц) и гигагерцах (ГГц). 1 МГц = миллион тактов в секунду.

Другой характеристикой процессора, влияющей на его производительность, является **разрядность процессора**.

Разрядность процессора определяется количеством двоичных разрядов, которые могут обрабатываться процессором одновременно. Часто уточняют разрядность процессора и пишут 64/36, что означает, что процессор имеет 64-разрядную шину данных и 36-разрядную шину адреса. Современный процессор Pentium 4 имеет разрядность 64/36, то есть одновременно процессор обрабатывает 64 бита, а адресное пространство составляет 68 719 476 736 байтов – 64 Гбайта.

Производительность процессора является его интегральной характеристикой, которая зависит от частоты процессора, его разрядности, а также особенности архитектуры (наличие кэш-памяти и др.). Производительность процессора нельзя вычислить, она определяется в процессе тестирования, по скорости выполнения процессором определенных операций в какой-либо программной среде.

Оперативная память. Оперативная память, предназначенная для хранения информации, изготавливается в виде модулей памяти. Модули памяти представляют собой пластины с рядами контактов, на которых размещаются БИС памяти. Модули памяти могут различаться между собой по размеру и количеству контактов (DIMM, DDR), быстродействию, информационной емкости и т.д.

Важнейшей характеристикой модулей оперативной памяти является быстродействие, которое зависит от максимально возможной частоты операций записи или считывания информации из ячеек памяти.

3. Базовая аппаратная конфигурация персонального компьютера

Персональный компьютер – универсальная техническая система.

Его *конфигурацию* (состав оборудования) можно гибко изменять по мере необходимости. Тем не менее, существует понятие *базовой конфигурации*, которую считают типовой. Понятие базовой конфигурации может меняться. В настоящее время в базовой конфигурации рассматривают четыре устройства:

- системный блок
- монитор

- клавиатура
- мышь

Системный блок представляет собой основной узел, внутри которого установлены наиболее важные компоненты. Устройства, находящиеся внутри системного блока, называют *внутренними*, а устройства, подключаемые к нему снаружи, называют *внешними*. *Внешние дополнительные устройства*, предназначенные для ввода, вывода и длительного хранения данных, также называют *периферийными*.

По внешнему виду системные блоки различаются формой корпуса. Корпуса персональных компьютеров выпускают в **горизонтальном** (*desktop*) и **вертикальном** (*tower*) исполнении. Корпуса, имеющие вертикальное исполнение, различают по габаритам:

- полноразмерный (*big tower*)
- среднеразмерный (*midi tower*)
- малоразмерный (*mini tower*).

Среди корпусов, имеющих горизонтальное исполнение, выделяют плоские и особо плоские (*slim*).

Кроме формы, для корпуса важен параметр, называемый *форм-фактором*. От него зависят требования к размещаемым устройствам.

Корпуса изготавливаются под унифицированные размеры материнских плат различного форм-фактора. Исходя из этой классификации, компьютерные корпуса бывают трех видов: **AT** (уже ушли в прошлое), **ATX** (современный вариант), **BTX** (новые модели). Форм-фактор корпуса должен быть обязательно согласован с форм-фактором главной (системной) платы компьютера, так называемой материнской платы.

Корпуса персональных компьютеров поставляются вместе с блоком питания и, таким образом, мощность блока питания также является одним из параметров корпуса. Для массовых моделей достаточной является мощность блока питания 250-300 Вт.

Монитор – устройство визуального представления данных.

Сейчас наиболее распространены плоские жидкокристаллические мониторы (ЖК).

В настоящее время комплектация монитора такова: корпус, блок питания, плата управления и экран. Информация на экран монитора выводится при помощи видеокарты.

Главные характеристики ЖК-мониторов:

- Разрешение – то есть размеры горизонтальной и вертикальной направленности;
- Размер точки – расстояние от одного пикселя до другого;
- Соотношение сторон экрана – ширины к высоте;
- Яркость – количество света, который передается через дисплей;
- Видимая диагональ – размер панели;
- Контрастность – соотношение яркости самой светлой и самой темной точек;
- Время отклика – минимальный промежуток времени, за который пиксель способен изменить свою яркость.

Достоинства жидкокристаллических мониторов:

- экономичность
- отсутствие электромагнитного излучения
- отсутствие мерцания
- легкость и удобство
- большая видимость экрана
- повышенная четкость изображения и цвета

Клавиатура – клавишное устройство управления персональным компьютером. Служит для ввода алфавитно-цифровых (знаковых) данных, а также команд управления. Комбинация монитора и клавиатуры обеспечивает простейший *интерфейс пользователя*.

Мышь – устройство управления манипуляторного типа. Перемещение мыши по плоской поверхности синхронизировано с перемещением графического объекта (указателя мыши) на экране монитора. Комбинация монитора и мыши обеспечивает наиболее современный *тип интерфейса пользователя*, который называется *графическим*.

4. Периферийные устройства персонального компьютера

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

По назначению периферийные устройства можно подразделить на:

- устройства ввода данных
- устройства вывода данных
- устройства хранения данных
- устройства обмена данными

Устройства ввода данных:

- устройства ввода знаковых данных – специальные клавиатуры (проводные и беспроводные);
- устройства командного управления – специальные манипуляторы (трекболы, пенмаусы, инфракрасные мыши, тачпады, джойстики, джойпады, геймпады, штурвально-педальные устройства и пр.);
- устройства ввода графических данных – сканеры, графические планшеты (дигитайзеры), цифровые фотокамеры.

Устройства вывода данных

Принтеры – матричные, струйные, лазерные, светодиодные.

Устройства хранения данных

- стримеры – накопители на магнитной ленте;
- накопители на съемных магнитных дисках;
- магнитооптические устройства;
- флэш-диски.

Устройства обмена данными

Модем – устройство, предназначенное для обмена информацией между удаленными компьютерами по каналам связи (**МО**дулятор + **ДЕМ**одулятор).

Контрольные вопросы

1. Что такое архитектура ЭВМ.
2. Какие существуют два типа архитектуры ЭВМ? В чем их отличие?
3. Укажите основные принципы Джона фон Неймана.
4. Перечислите основные устройства ЭВМ и дайте им краткую характеристику.
5. Что такое системная шина.
6. Назовите основные технические характеристики ЭВМ и как влияют на производительность компьютера?
7. Каковы назначение и характеристики микропроцессора?
8. Для чего служит память? Каких типов она бывает?
9. Зачем компьютеру память разных типов?
10. В чем заключается принцип открытой архитектуры?

Лекция 4.

КЛАССИФИКАЦИЯ И ТЕНДЕНЦИИ РАЗВИТИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1. Виды программного обеспечения.
2. Установка программ.
3. Правовая охрана программ и данных.
4. Перспективы развития программного обеспечения

1. Виды программного обеспечения.

Программа – это последовательность команд, которую выполняет компьютер в процессе обработки информации.

Программное обеспечение (англ. *Software* – «мягкое оборудование») – это совокупность программных и документальных средств для создания и эксплуатации систем обработки данных средствами вычислительной техники.

По функциональным возможностям программное обеспечение можно разделить на:

- *прикладные программы*
- *системные программы*
- *системы программирования*

Прикладное ПО – предназначено для выполнения конкретных задач пользователя.

Прикладное программное обеспечение

Программы	Пояснение	Примеры
Текстовые редакторы	Для ввода и редактирования текста	Блокнот, WordPad
Офисные пакеты <i>Microsoft Office</i> (коммерческий) <i>OpenOffice.org</i> (бесплатный) – крос-платформенное ПО, т.е. существуют его версии для разных ОС.	<i>Текстовый процессор</i> – стандартное средство для оформления офисных документов. <i>Табличный процессор</i> – программа для обработки табличных данных. <i>Компьютерная презентация</i> – набор слайдов, который предназначен для иллюстрации доклада или выступления. <i>СУБД</i> – это ПО для поиска информации в БД, а также для создания и изменения БД.	Microsoft Word OpenOffice.org Writer Microsoft Excel OpenOffice.org Calc Microsoft PowerPoint OpenOffice.org Impress Microsoft Access OpenOffice.org Base
Онлайн-офис	Специальные сайты, которые предоставля-	Google Docs

	ют основные возможности офисных пакетов: текстовый редактор, электронные таблицы, средства для создания презентаций.	OpenGoo
Графические редакторы	<p>Программы для создания и редактирования изображений (растровые и векторные).</p> <p>Растровые редакторы - обработка фото, подготовка изображений к печати, создание и редактирование изображений для web-сайтов.</p> <p>Векторные редакторы – для подготовки художественных и технических (схем, графиков) иллюстраций, логотипов, визиток, плакатов.</p> <p>Программы 3D-моделирования (3 измерения) – позволяют определить форму (геометрию) объектов сцены, задать материалы для объектов, установить источники света, создать анимацию с трехмерными объектами.</p>	<p>Paint, Paint.NET</p> <p>Adobe Photoshop</p> <p>GIMP</p> <p>Adobe Illustrator</p> <p>CorelDraw</p> <p>OpenOffice.org Draw</p> <p>3D Studio MAX</p> <p>Blender</p> <p>Компас-3D</p>
Последние версии растровых редакторов <i>Photoshop</i> , <i>GIMP</i> предоставляют пользователю и векторные инструменты.		
Настольно-издательские системы	Специальное ПО для подготовки к печати книг и журналов.	Microsoft Publisher PageMaker
Редакторы звука и видео	<p>Аудиоредакторы – это программы для редактирования звуковых файлов.</p> <p>Видеоредакторы предназначены для создания и редактирования цифрового видео.</p>	<p>Adobe Audition (звуковая станция)</p> <p>Audacity (редактор)</p> <p>Adobe Premiere</p> <p>Pinnacle Studio</p> <p>Kino</p>
Браузеры	ПО для работы в Интернете.	Internet Explorer Mozilla Firefox Google Chrome Safari Opera
Почтовые программы (почтовые клиенты)	ПО для работы с электронной почтой.	Почта Windows Microsoft Outlook TheBat
Мессенджеры	Программы для обмена мгновенными сообщениями (для общения в реальном времени).	ICQ Mail.ru Агент Skype Я.Онлайн Yahoo! Messenger

Системное ПО – это комплекс программ, которые обеспечивают эффективное управление аппаратными ресурсами вычислительной системы, такими как процессор, оперативная память, устройства ввода-вывода, сетевое

оборудование, выступая как «межслойный интерфейс», с одной стороны которого аппаратура, а с другой – приложения пользователя.

Системное программное обеспечение

Программы	Пояснение	Примеры
Операционные системы	Комплекс программ, обеспечивающих пользователю и прикладным программам удобный интерфейс (способ обмена информацией) с аппаратными средствами компьютера. В состав ОС входит специальная программа - командный процессор , которая запрашивает у пользователя команды и выполняет их.	Windows – семейство ОС с графическим многооконным интерфейсом. Unix – сетевая ОС Linux – бесплатная ОС MS-DOS
Драйверы	Специальные программы, которые обеспечивают управление работой устройств и согласование информационного обмена с другими устройствами, а также позволяют производить настройку некоторых их параметров. Каждому типу устройств соответствует свой драйвер.	
Утилиты		
Файловые менеджеры	Программы, обеспечивающие более комфортное общение пользователя с командами ОС	Проводник Windows Total Commander Far Manager
Программы диагностики	Инструменты для оценки, мониторинга и оптимизации ПК. ПО этой группы для предоставления информации об ОС, аппаратной составляющей ПК, их возможностях и работе в различных режимах.	AIDA64 - это мощная программа для диагностики всех параметров компьютера.
Антивирусные программы	Программы обнаружения компьютерных вирусов, для профилактики - предотвращения заражения файлов или ОС вредоносным кодом.	DrWeb NOD32 Kaspersky AVP Tool Avast Free Antivirus
Программы обслуживания дисков	Программы проверки целостности логической и физической структуры дисков, дефрагментация.	ScanDisk Norton Disk Doctor(NDD)
Архиваторы	Программы упаковки файлов для сокращения места на диске.	WinRar WinZip

Системы программирования – это программные средства для создания и отладки новых программ на конкретном языке программирования.

Системы программирования

Программы	Пояснение	Примеры
<p>Трансляторы – основа любой системы программирования</p> <p>Существует 2 типа трансляторов:</p> <p>Интерпретатор</p> <p>Компилятор</p>	<p>Программа, которая переводит в машинные коды тексты программ, написанных на языке высокого уровня.</p> <p>Анализирует текст программы по частям. Разобрав очередной фрагмент, он немедленно выполняет описанные в нем действия и переходит к обработке следующего фрагмента.</p> <p>Сразу переводит всю программу в машинный код и строит исполняемый файл, готовый к запуску.</p>	
Компоновщик	Программа, которая собирает разные части (модули) создаваемой программы и функции из стандартных библиотек в единый исполняемый файл.	
Отладчик	Программа для поиска ошибок в других программах.	
Профилировщик	Программа, позволяющая оценить время работы каждой процедуры и функции; используется для того, чтобы выяснить, какую именно процедуру нужно оптимизировать в первую очередь.	
Библиотеки стандартных подпрограмм	Набор готовых процедур и функций, которые можно вызывать из своей программы. Когда программа вызывает функцию из такой библиотеки, библиотека загружается в память и управление передается вызванной функции.	
Интегрированные среды разработки программ (входят текстовый редактор для набора текста программ, транслятор, компоновщик, отладчик, профилировщик)	Объектно-ориентированные языки программирования.	Microsoft Visual Studio Delphi Visual Basic

2. Инсталляция программ

Процесс установки программ на ПК называется **инсталляция** (*инсталлировать*, от англ. *install* – установить).

Для установки программы необходимо иметь **дистрибутив** (установочный пакет, от англ. *Distribute* – распространять), т.е. набор файлов и папок, в которых хранится весь комплекс программ в сжатом виде. Один из файлов дистрибутива является установочным, т.е. при его запуске начинается процесс установки; как правило, он имеет название Setup.exe или Install.exe.

Обычно *установка включает несколько этапов* (некоторые из них могут отсутствовать):

- выбор компонентов программы, которые пользователь хочет установить;
- просмотр лицензионного соглашения (договора о возможности использования программы);
- ввод ключа (серийного номера) программы;
- определение каталога, в котором нужно разместить файлы программы;
- распаковка и копирование файлов на жесткий диск компьютера;
- настройка программы с помощью файлов конфигурации (или запись настроек в системный реестр в ОС *Windows*);
- создание ярлыков для запуска программы в меню и/или на *Рабочем столе*.

В ОС *Windows* для установки программ используется служба *Windows Installer*, которая работает с установочными пакетами – файлами в формате **.msi**. Дистрибутив также может представлять собой программу (файл с расширением **.exe**), которая содержит все необходимые данные и при запуске «ведёт» пользователя через все этапы установки.

3. Правовая охрана программ и данных

По законам большинства стран компьютерные программы и данные охраняются *авторским правом*. Это значит, что автор (или правообладатель, например, фирма, в которой работает автор) может ограничивать распространение и использование программы.

В Конституции Российской Федерации записано, что «*интеллектуальная собственность охраняется законом*» (ст. 41 ч. 1). **Интеллектуальная собственность** – это права на результаты творческой деятельности человека.

Эти права детально определены в Гражданском кодексе РФ (часть IV, «Права на результаты интеллектуальной деятельности и средства индивидуализации»).

Авторские права **распространяются** на:

- *программы для компьютеров* (включая подготовительные материалы, а также звук, графику и видео, которые получаются с помощью программы);
- *базы данных* (массивы данных, специально организованные для поиска и обработки с помощью компьютеров).

Не охраняются авторским правом

- *алгоритмы и языки программирования;*
- *идеи и принципы,* лежащие в основе программ, баз данных, интерфейса;
- *официальные документы.*

Согласно российским законам об авторском праве, автор – это физическое лицо (не организация).

Авторское право

- возникает «в силу создания» продукта и не требует формальной регистрации, хотя при желании автор может зарегистрировать программу в государственных органах;
- обозначается знаком ©, после которого записывается фамилия автора и год первого выпуска программы, например, © *Иванов, 2008*;
- действует в течение жизни и 70 лет после смерти автора;
- передается по наследству.

Автор получает **личные права**:

- право авторства (право считаться автором);
 - право на имя (право выпускать программу под своим именем, псевдонимом или анонимно);
 - право на неприкосновенность программы и ее названия;
- и **имущественные права**: осуществлять или разрешать
- выпуск программы в свет;
 - копирование в любой форме;
 - распространение;
 - изменение (в т.ч. перевод на другой язык).

Серьезные нарушения авторских прав могут попасть под действие Уголовного кодекса РФ (ст. 146, «Нарушение авторских и смежных прав»). Уголовная ответственность наступает при крупном ущербе (более 50 000 руб.). Присвоение авторства (*плагиат*) наказывается лишением свободы на срок до 6 месяцев. В случаях незаконного использования, а также приобретения и хранения объектов авторского права (например, дисков с нелегальными программами) в целях сбыта срок лишения свободы может достигать 5 лет (при особо крупном ущербе).

Типы лицензий на использование ПО

Право на использование программы дает документ (договор), который называют **лицензией** (лат. *litentia*) или *лицензионным соглашением*. Это соглашение между правообладателем и пользователем, где четко определены права и обязанности сторон. Как правило, в соответствии с лицензией пользователь без дополнительного разрешения автора может:

- установить программу на 1 компьютер (или так, как указано в договоре);
- вносить изменения, необходимые для работы программы на компьютере пользователя; исправлять явные ошибки;
- изготовить копию, чтобы можно было восстановить программу в случае сбоя;
- передать программу другому лицу вместе с лицензией.

Программы, которые получены и используются в соответствии с законом, называют *лицензионными*. Если же при создании копии были нарушены авторские права, ее называют *контрафактной* или *пиратской*.

По **типу лицензий** можно разделить ПО на 4 типа:

- *коммерческое (лицензионное)* – приобретается через официальную торговую сеть или поставляется вместе с ПК;
- *условно бесплатное* (англ. *shareware*) – является рекламой коммерческих программ, как правило имеет ограниченные функциональные возможности или ограниченный срок работы (месяц);
- *бесплатное* (англ. *freeware*) – распространяется в основном через Интернет;
- *свободное ПО* (англ. *open source* – ПО с открытым кодом).

Лицензия на программное обеспечение – это правовой инструмент, определяющий использование и распространение программного обеспечения, защищенного авторским правом. Обычно лицензия на программное обеспечение разрешает получателю использовать одну или несколько копий программы, причем без лицензии такое использование рассматривалось бы в рамках закона как нарушение авторских прав издателя. По сути, лицензия выступает гарантией того, что издатель ПО, которому принадлежат исключительные права на программу, не подаст в суд на того, кто ею пользуется.

Свободное программное обеспечение (СПО) – программное обеспечение, пользователи которого имеют права («свободы») на его неограниченную установку, запуск, а также свободное использование, изучение, распространение и изменение (совершенствование), и распространение копий и результатов изменения. Если на программное обеспечение есть исключительные права, то свободы объявляются при помощи свободных лицензий.

Как и бесплатное и бесплатно распространяемое программное обеспечение, СПО можно использовать и обычно получить бесплатно (но конкретный распространитель может взимать плату за получение у него копий, за каналы доставки, носители – компакт-диски или дополнительные сервисные услуги). Однако **бесплатное ПО** обычно распространяется в исполнимом виде без исходных кодов и является проприетарным ПО, а чтобы ПО было свободным, получателям должны быть доступны его исходные коды, из которых можно создавать исполнимые файлы, вместе с соответствующими лицензиями. Из-за того, что словом «продажа» называют и продажу оригинала, и распространение копий за деньги, и платное лицензирование (обычно на несвободных условиях), некоторые все же считают свободное ПО подмножеством бесплатного.

Проприетарное программное обеспечение – программное обеспечение, являющееся частной собственностью авторов или правообладателей и не удовлетворяющее критериям свободного ПО (наличия открытого программного кода недостаточно). Правообладатель проприетарного ПО сохраняет за собой монополию на его использование, копирование и модификацию, полностью или в существенных моментах. Обычно проприетарным называют любое несвободное ПО, включая *полусвободное*.

4. Перспективы развития программного обеспечения

Ориентация на взаимодействие

Одновременно с развитием сетевых технологий ПО становится все больше ориентированным не на индивидуальную работу или развлечение, а на взаимодействие – рабочее или социальное. Все больше возникает интернет-порталов и социальных сетей, в которых люди находят себе группы по интересам, сами технологии взаимодействия непрерывно совершенствуются.

Когда-то чудом было получить письмо через несколько минут после его отправки, сегодня уже не редкость сидеть за компьютером, общаясь с несколькими людьми голосом и визуально, видеть их на экране компьютера, знать, что они видят тебя.

Компьютерные лаборатории ведут активные разработки технологий и программного обеспечения, позволяющие передавать через сетевые коммуникации вкус, запах, кинестетические ощущения.

Интеграция с сетью

Компьютерное программное обеспечение все больше и больше интегрируется с локальными и глобальными сетями. Все больше приложений, которые мы привыкли видеть в качестве настольных, становятся сетевыми.

Показательным примером может служить служба документов Google, позволяющая выполнять полноценные офисные операции с документами.

Например, можно создавать документы, электронные таблицы и презентации в режиме подключения к сети, не устанавливая офисные пакеты на свой компьютер.

Компьютеры без операционной системы

Развитием идеи интеграции программного обеспечения с сетью будут компьютеры, состоящие из одного «железа» и небольшой программы подключения к сети и загрузки сетевых приложений. Необходимое ПО можно получить из Интернета после включения компьютера в сеть, например, рабочий стол, приложения, хранилища для файлов. Документы Google – это только первый шаг на пути тотальной сетевой интеграции.

Распределенность

Развитие и расширение каналов сетевых коммуникаций оказало влияние и на технологии создания приложений различных классов: все чаще на компь-

ютере пользователя оказывается только каркас, оболочка, а все необходимые вычисления производятся где-то в сети. Причем тенденцией сегодняшнего дня является сокрытие от пользователя, средств выполнения вычислений и мест хранения «строительных блоков» запущенного приложения.

Автономизация и самообучение, повышение интеллектуальности

ПО

Создаваемое сегодня ПО все чаще обретает черты интеллектуальности. Оно не просто решает наши задачи, оно пытается угадать, что нам удобно, проанализировать, что мы делаем чаще, и предложить нам сформированные инструменты для этих действий или готовые решения. Ведутся активные разработки в области создания автономных компьютерных информационных систем, способных длительное время функционировать без вмешательства человека, самообучающихся, самообслуживающихся и самостоятельно определяющих свои цели.

Интеграция с бытовой техникой

Все больше ПО влияет не только на наши работу и развлечения, но и на повседневную жизнь: ПО «умного дома» позволяет полностью переложить многие операции на плечи компьютеров и связанной с ними аппаратуры. Все больше электронных устройств имеют не просто автоматизированную, но интеллектуально-компьютеризированную начинку. По мере удешевления микропроцессоров все больше усилий вкладывается в создание ПО для них, способного немедленно объединить недавно купленную бытовую технику с тем семейством приспособлений, которое уже имеется в наличии.

Новые виды пользовательского интерфейса

Одна из тенденций в развитии ПО последних лет состоит в том, чтобы отказаться от сложившегося графического пользовательского интерфейса (рабочий стол, контекстные меню, главное меню в программах, стандартные кнопки окон) и найти какое-то новое решение. Сегодня еще нельзя сказать) в правильном ли направлении идут эти поиски, но ведь и компьютерная технология не стоит на месте: вполне возможно, лет через пять эти вопросы перестанут быть актуальными, и разрабатываться будут новые интерфейсы к трехмерному рабочему столу, проецируемому в пространство пакетом голографических лазеров.

Контрольные вопросы

1. Назовите основные уровни программного обеспечения. Каков порядок их взаимодействия?
2. Дайте классификацию видов программного обеспечения.
3. Какие классы программного обеспечения по виду лицензирования вы знаете?
4. В чем разница между свободным, открытым, коммерческим и проприетарным программным обеспечением?
5. Что такое «опасное программное обеспечение»?
6. Перечислите известные вам классы прикладного программного обеспечения.
7. Идентичны ли функционально программы, входящие в состав пакетов Microsoft Office и OpenOffice.org?
8. Какие программы для работы с сервисами Интернета вам известны?
9. Как создавать документы и электронные таблицы в распространенных «офисных» форматах, не устанавливая на компьютер офисный пакет?
10. Каковы перспективы развития программного обеспечения?

Лекция 5.

ОПЕРАЦИОННЫЕ СИСТЕМЫ СЕМЕЙСТВА WINDOWS

1. Особенности операционных систем семейства Windows.
2. Графический интерфейс.
3. Файловая система.

1. Особенности операционных систем семейства Windows

ОС семейства Windows характеризуются следующими особенностями:

Единый пользовательский интерфейс

Средства общения ПК с пользователем для всех Windows-программ в значительной степени унифицированы, поэтому не требуется для каждой программы изучать новые правила взаимодействия и управления. Нет необходимости запоминать имена программ и документов, так как они изображаются на экране специальными значками.

Многозадачность

Обеспечивается возможность одновременного выполнения нескольких программ.

Совместимость с DOS-приложениями

Для запуска программ, работающих под управлением DOS, нет необходимости выходить из Windows, их работу можно просматривать в одном из окон.

Средства обмена данными

Для организации обмена данными между приложениями используются следующие способы:

- **буфер обмена** – одна программа помещает туда данные, а другая их использует;
- **динамический обмен данными** – одна программа использует данные, созданные другой, при этом копия данных сохраняет привязку к исходной программе;
- **механизм связи и внедрение объектов** – приложение, использующее данные, может запустить исходную программу для редактирования данных.

Поддержка масштабируемых шрифтов

Дает возможность строить шрифты любых нужных размеров.

Удобство поддержки устройств

Вся работа прикладных программ с внешними устройствами осуществляется через посредство Windows, для подключения к ПК любого нового устройства достаточно установить драйвер этого устройства, после чего каждая из программ может обращаться к этому устройству.

Поддержка мультимедиа

При подключении некоторых дополнительных устройств Windows может выполнять обработку звуковой информации и видеоизображения. Это открывает большие возможности для обучающих, игровых и других прикладных программ, позволяет делать общение с ПК более легким и приятным.

2. Графический интерфейс

Для упрощения работы пользователя в состав современных ОС входят программные модули, создающие *графический пользовательский интерфейс*.

Рабочий стол – это графическая среда, на которой отображаются *объекты* и *элементы управления* Windows.

Значки (*пиктограммы*) – это графическое представление объектов Windows.

Панель задач – основной элемент управления ОС.

Одним из самых важных объектов ОС Windows является **окно**.

Типы окон:

- окно папки;
- окно программы;
- окно документа;
- диалоговое окно;
- окно справочной системы.

Хотя содержимое каждого окна индивидуально, у всех окон есть одни и те же **основные элементы**:

- строка заголовка с кнопками управления окном;
- строка меню;

- рабочая область;
- полосы прокрутки;
- строка состояния.

Меню представляет собой список взаимосвязанных команд, используемых для выполнения задач при работе с ОС Windows или приложениями. Команды меню объединены в логические группы.

Существуют следующие виды меню:

Главное

Контекстное

Системное

Программное (строка меню)

Особенности программного меню:

1. если команда в меню выделена тусклым цветом, использовать ее в данный момент нельзя;
2. если рядом с названием пункта стоит значок ►, то при выборе данной команды, раскроется соответствующее подменю;
3. если рядом с командой меню стоит ..., то при выборе данной команды откроется соответствующее диалоговое окно;
4. если справа от команды меню указана комбинация клавиш, то указанную команду можно заменить этой комбинацией клавиш;
5. группа команд со схожими функциями, отделяются горизонтальными линиями;
6. если перед названием пункта стоит галочка «✓», то это означает, что режим, указываемый данной командой, включен;
7. если перед названием пункта меню стоит символ «●», то это означает, что из нескольких альтернативных режимов, сейчас включен режим, указываемой данной командой.

3. Файловая система

Способ хранения файлов на дисках компьютера называется **файловой системой**.

В процессе работы компьютера происходит обмен файлами между устройствами. Разные ОС используют различные файловые системы, т.е.

различные способы организации, хранения и именования данных на носителях информации.

В ОС имеются программные модули, управляющие файловой системой.

Минимальным адресуемым элементом носителя информации является кластер, который может включать в себя несколько секторов (объем сектора составляет 512 байтов). Файловая система организует кластеры в файлы и каталоги.

Файлом называется поименованная совокупность данных, хранящихся на внутреннем или внешнем носителе информации.

Имя файла состоит из **имени** и **расширения**, служащего для определения типа данных, хранящихся в файле. Имя файла и его расширение разделяются точкой.

Для **характеристики файла** используются следующие параметры:

- *полное имя файла;*
- *объем файла в байтах;*
- *дата создания файла;*
- *время создания файла;*

специальные атрибуты файла:

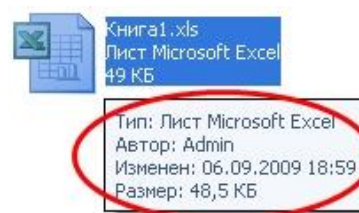
- R (Read only) – *только для чтения;*
- H (Hidden) – *скрытый файл;*
- S (System) – *системный файл;*
- A (Archive) – *архивированный файл.*

Кроме имени и расширения файл характеризует значок программы.

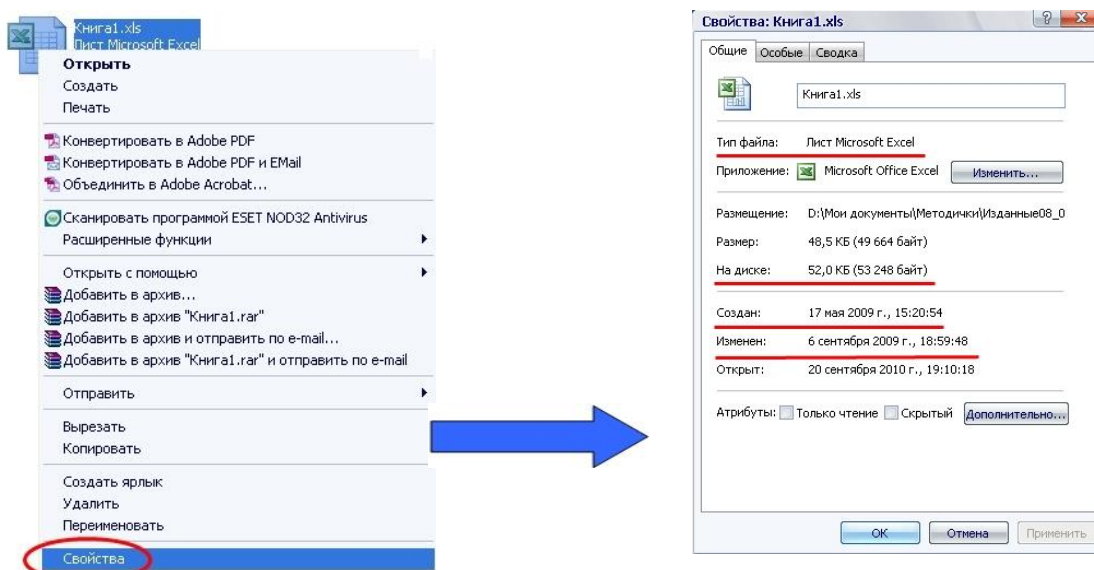
Примеры значков файлов:



Параметры файла можно просмотреть следующим образом: выделив объект щелчком левой кнопки мыши и наведя на него указатель мыши,



появится всплывающая подсказка, отображающая основные параметры файла, либо



Файловая система позволяет организовывать файлы в иерархические древовидные каталоги (папки). Каталог самого верхнего уровня называется **корневым каталогом** (обозначается именем логического источника данных – латинской буквой с двоеточием). Остальные каталоги делятся на **системные** и **пользовательские**. Системные каталоги имеют служебные имена, назначенные ОС при установке. Имена пользовательских каталогов задаются пользователем по его усмотрению.

Пользователь может передвигаться по каталогам с помощью **программ навигации** (Проводник) или с помощью специальных команд ОС.

Последовательность имен каталогов, проходимых от некоторого каталога до нужного файла, называется **путем** к этому файлу. Имена каталогов в этой последовательности принято разделять знаком \ .

Различают **абсолютное** (полное) и **относительное** имя файла. **Абсолютное имя** складывается из пути к файлу от корневого каталога и имени файла. **Относительное имя** – из пути к файлу из текущего каталога и имени файла.

D:\МОИ ДОКУМЕНТЫ\ИНФОРМАТИКА\ЗАДАНИЯ\Вариант1.doc

Абсолютное имя

Относительное имя

Иерархическая структура, в виде которой операционная система отображает файлы и папки диска, называется **файловой структурой**.

К основным операциям с файловой структурой относятся:

- навигация по файловой структуре;
- запуск программ и открытие документов;
- создание папок;
- копирование файлов и папок;
- перемещение файлов и папок;
- удаление файлов и папок;
- переименование файлов и папок;
- создание ярлыков.

Для групповых операций с файлами (поиск, удаление, перемещение) используется **маски** имен файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

***** (*звездочка*) – служит для замены любой последовательности символов

? (*вопросительный знак*) – служит для замены одного символа. В шаблоне может быть использовано несколько таких символов.

| (*вертикальная черта*) – используется в качестве связки ИЛИ.

Примеры:

Маска для выделения всех файлов в текущем каталоге: *.*

Маска для выделения файлов с расширением .doc: *.doc

Маска для выделения текстовых файлов с односимвольным именем ?.txt

Контрольные вопросы

1. Дайте характеристику операционным системам нового поколения.
2. Назовите особенности операционных систем семейства Windows.
3. Операционная система Windows – структура, пользовательский интерфейс.
4. В чем состоят функции файловой системы?
5. Какие операции с файлами удобно проводить с использованием шаблонов?
6. Какими способами может быть организован доступ к файлу?
7. Что такое окно? Какие существуют разновидности окон в Windows?
8. Для чего предназначен буфер обмена?
9. Какие виды меню вы знаете? Укажите особенности программного меню.
10. Какие вы знаете операционные системы, не входящие в семейство Windows?

Лекция 6.

ОФИСНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. *Текстовый процессор OpenOffice.org Writer. Технологии обработки текстовой информации.*
2. *Табличный процессор OpenOffice.org Calc. Обработка данных средствами электронных таблиц.*
3. *OpenOffice.org Impress. Программные технологии создания мультимедиа-презентаций.*

OpenOffice.org (OOo) – это свободный международный бесплатный набор офисных программ с открытым исходным кодом, распространяемых в составе Пакета свободного программного обеспечения для образовательных учреждений Российской Федерации. Он полностью совместим с офисным пакетом Microsoft Office, что немаловажно при крупномасштабном переходе на новый офисный пакет.

1. Текстовый процессор OpenOffice.org Writer.

Технологии обработки текстовой информации.

Текстовый процессор Writer является одним из компонентов офисного пакета **OpenOffice.org**. Приложение **OpenOffice.org Writer** позволяет создавать текстовые документы практически любой сложности, содержащие графические объекты, таблицы и диаграммы.

Окно программы (рис. 1) содержит следующие элементы:

Заголовок окна – слева в строке заголовка находится значок Writer, имя документа *Безымянный1*, имя программы OpenOffice.org Writer и кнопки управления окном.

Строка меню находится под строкой заголовка. Соответствующий пункт меню раскрывается щелчком мыши по нему.

Панели инструментов находятся ниже строки меню. Панели *Стандартная* и *Форматирование* отображаются на экране по умолчанию. Остальные панели при необходимости могут быть открыты командами **Вид – Панели инструментов**. Внутри панелей инструментов собраны различные элемен-

ты управления, которые представляют наиболее часто выполняемые операции при работе с документами. Панели инструментов обеспечивают быстрый и более простой доступ к важнейшим функциям текстового процессора.

Рабочая область – это область экрана, где отображается создаваемый документ и производится его редактирование. Сверху и слева находятся горизонтальная и вертикальная линейки. Справа и снизу – полосы прокрутки.

Строка состояния находится внизу окна программы. Это – информационная строка, где указывается различная информация о самом документе и режимах программы – число страниц, текущее положение страницы и др.

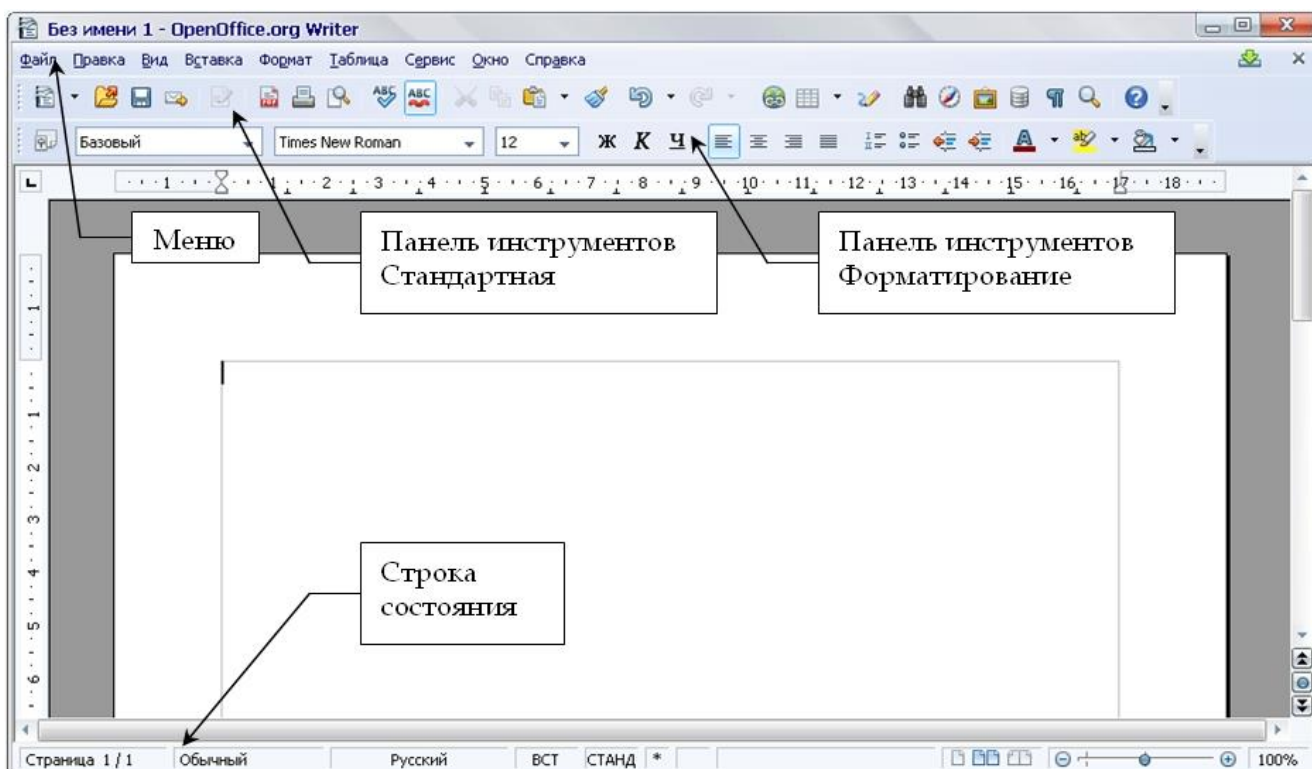


Рис. 1. Рабочее окно программы Writer

Writer имеет несколько режимов отображения документа:

- *Разметка печати*
- *Режим веб-страницы*
- *На весь экран*
- *Масштаб*

Доступ к выбору этих режимов осуществляется с помощью меню **Вид**. Только режим *Масштаб* имеет свое подменю.

Выбор параметров страниц

Разметка страницы служит для создания печатных документов.

В начале работы над документом необходимо задать значения параметров страницы (меню **Формат – Страница**).

Существуют форматы страниц:

- А4 (21 × 29,7 см) – размер стандартного листа бумаги для принтера (рефераты, курсовые, дипломы, заявления);
- А3 (29,7 × 42 см) – размер в два раза больше стандартного листа (объявления, плакаты);
- А5 (14,8 × 21 см) – размер в два раза меньше стандартного листа (письма).

Существуют две возможные ориентации страницы:

- *книжная;*
- *альбомная.*

На странице можно установить требуемые **размеры полей**, которые определяют расстояние от краев страницы до границы текста.

Поля: слева – 3 см;
 справа – 1,5 см;
 сверху – 2 см;
 снизу – 2 см.

Для вывода на каждой странице документа одинакового текста удобно использовать верхний и нижний колонтитул.

Колонтитулами называют области, расположенные в верхнем и нижнем полях страницы. Информация, помещенная в колонтитул, будет печататься на каждой странице документа.

В случае если необходимо, чтобы отдельные страницы документа имели свои параметры оформления, документ разбивается на разделы (меню **Вставка – Разрыв ... – Разрыв страницы**). Каждый раздел документа может иметь свой колонтитул.

Колонтитулом может быть: **Вставка – Поля – Время (Дата, Тема, Заголовок, Автор)**.

Для нумерации страниц используется меню **Вставка – Поля – Номер страницы**.

Работа с текстом

При вводе текста с клавиатуры по достижении конца строки слово авто-

матически переносится на следующую строку. Каждый абзац заканчивается нажатием клавиши **ENTER**.

Для удаления символов используются клавиши **Delete** и **Backspace**.

Текстовый процессор по умолчанию записывает текст в режиме **Вставка**. При нажатии клавиши **INSERT** программа переходит в режим **Замены**. Переключение между режимами осуществляется нажатием клавиши **INSERT** или щелчком мыши в строке состояния в области **ВСТ (ЗАМ)**.

На панели инструментов *Стандартная* имеется кнопка **Непечатаемые символы** ¶, которая включает режим отображения на экране непечатаемых символов.

Непечатаемые символы – это специальные символы, которые используются при обработке документа. К непечатаемым символам относятся: *маркер конца абзаца* (знак ¶), *пробел* (знак ·), *символ табуляции* (знак →) и др.

Проверка орфографии

Writer обеспечивает проверку орфографии, которая может быть вызвана двумя способами.



Автопроверка орфографии проверяет каждое слово при его вводе.



Для индивидуальной проверки орфографии в документе (или выделенного текста) нужно щелкнуть по кнопке **Проверка орфографии**, открывается окно **Проверка грамматики**, если найдено любое ошибочное слово.

Если опции автоматической проверки правописания включены, то текстовый процессор сразу указывает на ошибки набора текста и подчеркивает красной волнистой линией слова с орфографическими ошибками и зеленой волнистой линией грамматические ошибки.

При вызове контекстного меню на подчеркнутом слове из раскрывающегося списка можно выбрать один из предложенных вариантов написания слова. В случае затруднения подбора слов в предложении можно использовать словарь синонимов (*тезаурус*), который вызывается через контекстное меню.

Тезаурус – это особая разновидность словаря, в котором слова сгруппированы по определенным смысловым признакам, *например*, синонимам или антонимам.

Перенос слов по слогам задается в меню:

Формат – Абзац – Положение на странице – Автоматический перенос.

Вставка специальных символов в тексте

Специальный символ – это символ, которого нет на стандартной клавиатуре.

Например, символы © ¾ ® ÷ ± ♀ ♂ ☺ являются специальными.

Чтобы вставить в текст специальный символ следует:

1. Выбрать **Вставка – Специальные символы...** для отображения диалогового окна **Выбор символа** (рис. 2).
2. Выделить необходимые символы в порядке их следования в тексте и нажать кнопку **Ок**. Выбранные символы отображаются в нижнем левом углу окна. После выделения символа он показывается в правом нижнем углу вместе с его числовым кодом.

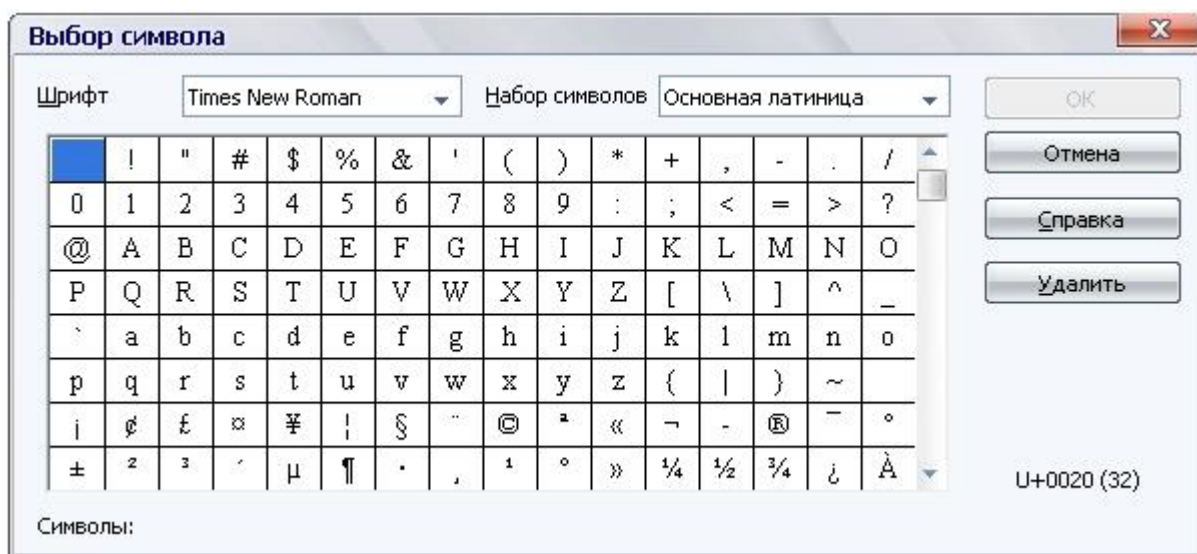


Рис. 2. Диалоговое окно Выбор символа

Разные шрифты имеют разные специальные символы. Если не найден конкретный специальный символ, следует выбрать другой Шрифт.

Маркеры и нумерация

Маркеры и нумерация применяются для размещения в документе различных перечней. Существуют маркеры и нумерация различных типов:

- нумерованные списки (тип нумерации);
- маркированные списки (маркеры);
- многоуровневые списки (структура).

Для преобразования существующего текста в нумерованный или маркированный список необходимо выделить этот текст, выбрать команду **Формат – Маркеры и нумерация** и щелкнуть на вкладке **Тип нумерации** или соот-

ответственно **Маркеры**. Вкладка **Структура** позволяет задать специальный список, содержащий до десяти уровней пунктов, нумерованных или маркированных отдельно. На вкладке **Структура** выбрать требуемый тип многоуровневого списка (рис. 3).

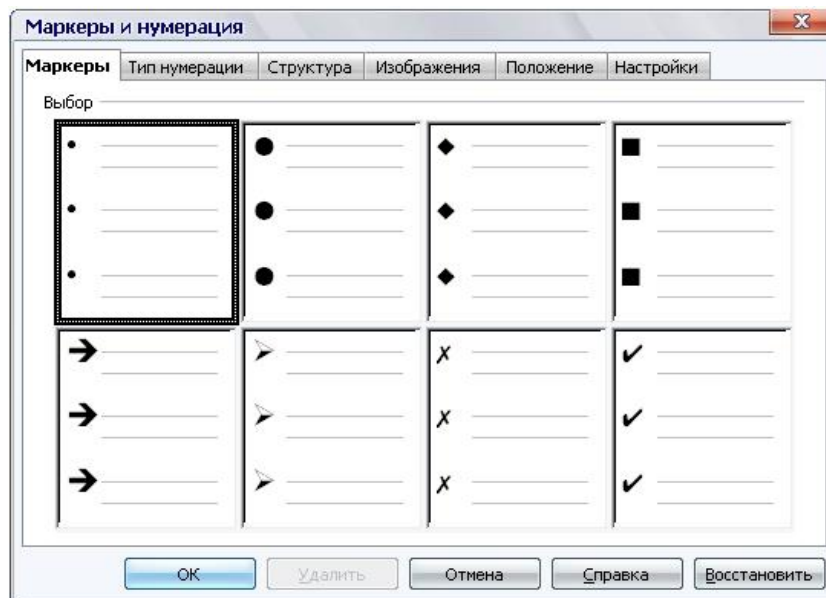


Рис. 3. Диалоговое окно *Маркеры и нумерация*

Форматирование символов

Символы (буквы, цифры, знаки пунктуации, различные символы) являются основными объектами, из которых состоит текст документа. Символы можно форматировать.

Среди основных свойств можно выделить следующие: *шрифт*, *размер*, *начертание* и *цвет*.

Диалоговое окно, вызываемое командой **Формат – Символы**, позволит задать параметры символов, а также применить к символам большое количество форматов, можно используя кнопки на Панели форматирования.

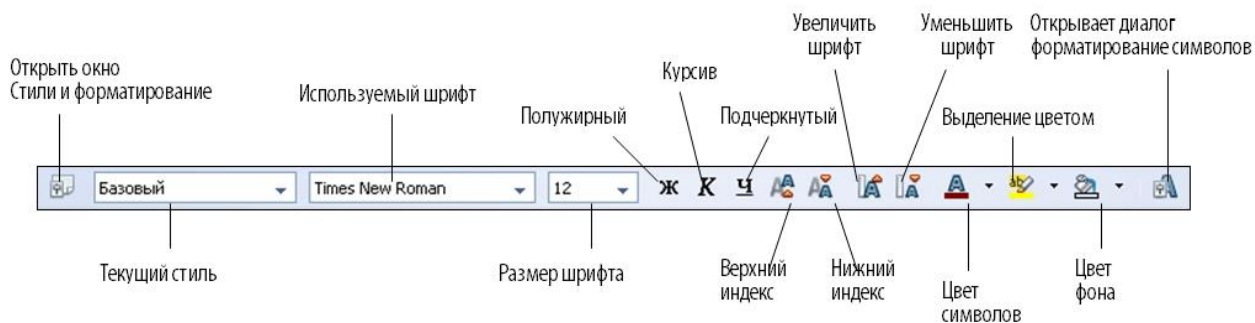


Рис. 4. Панель форматирования символов

Единицей измерения размера шрифта является пункт (1 пт = 0,376 мм). Размер шрифта в полиграфии называют **кеглем**. Размеры шрифтов можно изменять в больших пределах. Ниже приведены примеры представления текста при различных размерах шрифта.

Шрифт размером 20 пт.

Шрифт размером 16 пт.

Шрифт размером 12 пт.

Шрифт размером 8 пт.

Кроме обычного начертания символов, применяют *курсив*, **полужирный**, **полужирный курсив**. Можно установить дополнительные параметры форматирования символов: подчеркивание линиями различных типов, изменение вида символов (приподнятый, утопленный), изменение расстояния между символами (*разреженный*, *уплотненный*) и др.

Если планируется цветная печать документа, то для различных символов целесообразно задать различные цвета, выбранные из предлагаемой палитры текстовым редактором.

Форматирование абзацев

Абзац является одним из основных объектов текстового документа. В компьютерных документах абзацем считается любой текст, заканчивающийся управляющим символом конца абзаца. Ввод конца абзаца обеспечивается нажатием клавиши **Enter** и отображается символом ¶, если включен режим отображения непечатаемых символов. Этот режим включается и отключается нажатием кнопки на панели инструментов *Стандартная*.

Важным параметром при наборе текста является его *выравнивание*. Выравнивание отражает расположение текста относительно границ полей страницы. Существует четыре способа выравнивания абзацев:

- по левому краю;
- по центру;
- по правому краю;
- по ширине.

Для выравнивания абзаца необходимо сделать установки на вкладке **Выравнивание** в диалоговом окне **Абзац (Формат – Абзац)** или изменить выравнивание абзаца с помощью кнопок на панели инструментов *Форматирование*.

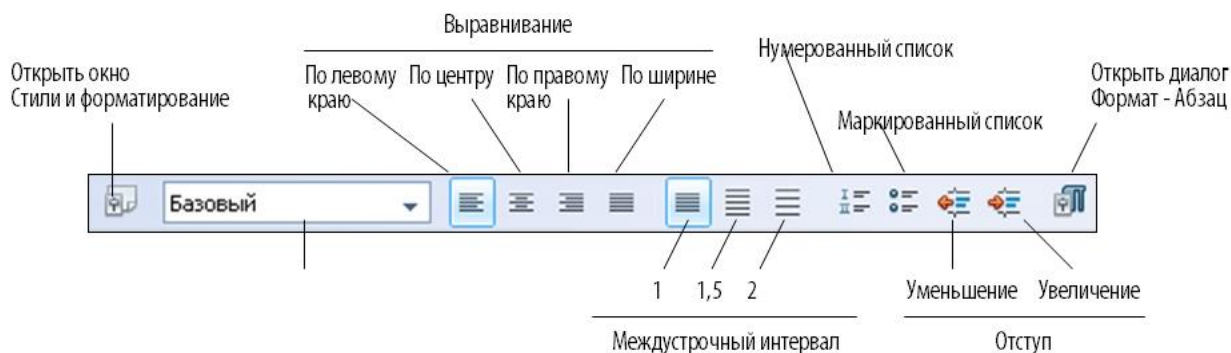


Рис. 5. Панель форматирования абзацев

Чаще всего абзац начинается *отступом первой строки*. Отступ может быть различных типов:

- положительный отступ (положительное значение), когда первая строка начинается правее всех остальных строк абзаца, применяется в обычном тексте;
- отрицательный отступ (отрицательное значение), когда первая строка выходит влево относительно остальных строк абзаца, применяется в словарях и определениях;
- нулевой отступ, применяется для абзацев, выровненных по центру.

Для установки типа отступа первой строки абзаца понадобится установить необходимое значение в списке **Первая строка** и задать конкретное числовое значение отступа с помощью счетчика справа.

Весь абзац может иметь *отступы слева и справа*, которые отмеряются от границ полей страницы. Для задания отступа абзаца от границ полей страницы следует выбрать нужное значение отступа с помощью счетчиков **Отступ слева** и **Отступ справа**.

Расстояние между строками документа можно изменять, задавая различные значения *междустрочных интервалов* (одинарный, двойной и т. д). Для визуального отделения абзацев друг от друга или одного абзаца от остальных нужно устанавливать увеличенные интервалы между абзацами. Междустрочный интервал выбирают с помощью раскрывающегося списка **Междустрочный**, а интервал перед (после) абзаца – с помощью счетчиков **Интервал перед** и **после**.

Вкладка **Положение на странице** позволяет установить требуемое распределение абзацев по страницам, то есть запретить разрывать абзац между

страницами, оставлять на странице первую или последнюю (висячую) строку и т.д.

Работа с таблицами

Для работы с таблицами в текстовом процессоре имеется специальное меню **Таблица**. С помощью этого меню можно создать таблицу (**Таблица – Вставить таблицу**) и изменить ее размеры, добавляя и удаляя строки и столбцы (пункты **Удалить**, **Вставить** в меню **Таблица**).

По умолчанию в таблицу вставляется одна строка или столбец. Если перед добавлением выделено несколько строк или столбцов, то в таблицу вставляется столько строк и столбцов, сколько было выделено.

OOo Writer позволяет создавать сложные структуры таблиц, в частности с помощью команд **Таблица – Разбить ячейки (Объединить ячейки)**.

Текст в ячейках таблицы форматируется так же, как и абзацы документа.

Для расположения текста в ячейке вертикально используются команды **Формат – Символы – Положение**.

Некоторые пункты меню **Таблица** вынесены на панель инструментов **Таблица**.

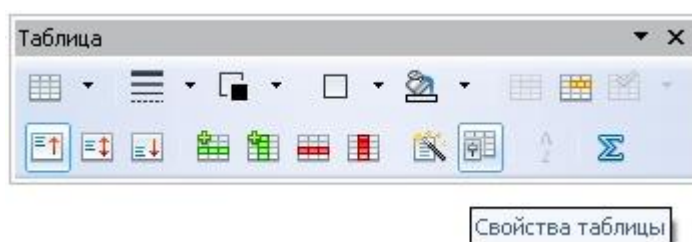


Рис. 6. Панель инструментов **Таблица**

Дополнительные элементы оформления

Текстовый процессор **OOo Writer** предоставляет большие возможности по оформлению документа. Это может быть многоколонный набор текста, названия, сноски, рамки, гиперссылки.

Многоколонный набор (Формат – Колонки) используется при подготовке буклетов и газетных полос, **названия** применяются для создания подписей рисунков, таблиц и формул, **рамки** используются для подготовки веб-документов. В **сноски** и **примечания** (меню **Вставка**) помещаются различные поясняющие комментарии.

Имеются два типа сносок – **обычные** и **концевые**. **Обычная сноска** располагается внизу страницы, а **концевая** – в конце раздела или документа. Один

документ может содержать и обычные, и концевые сноски. Например, обычные сноски можно использовать для разъяснения вводимых терминов, а концевые – для ссылки на первоисточники.

Текстовый процессор **OOo Writer** поддерживает технологию объектного связывания, при которой в текстовый документ можно вставлять документы, созданные другими приложениями. Объектами могут быть листы рабочей книги табличного процессора, слайды презентации, диаграммы, формулы и др. Для вставки объекта используется меню **Вставка – Объект**.

Стили

Стилем называется набор параметров форматирования объекта, имеющий имя.

Стили применяются для отдельных элементов текста, таблиц, а также для всего документа. Применение стиля позволяет одним действием изменить сразу все оформление документа.

Стиль знака задает форматирование выделенного фрагмента текста внутри абзаца, определяя такие параметры текста, как шрифт, размер, начертание.

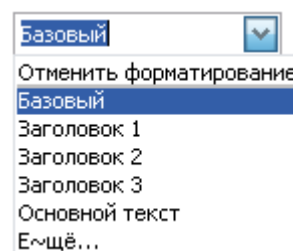
Стиль абзаца полностью определяет его внешний вид, т.е. выравнивание текста, междустрочный интервал и границы; он может также включать форматирование знаков.

Стиль таблицы задает вид границ, заливку, выравнивание текста и шрифты.

Некоторые специальные элементы текста, такие, как ссылки, названия, заголовки, цитаты, сноски имеют **стандартные** встроенные **стили** оформления. Стандартные стили нельзя удалять, но их можно изменять.

Помимо стандартных стилей пользователь может создавать и применять свои собственные стили. Новые созданные стили называются **пользовательскими**. Пользовательские стили в отличие от стандартных встроенных стилей можно удалять.

Для применения стиля на панели инструментов **Форматирование** имеется кнопка со списком перечня стилей:



Структура документа

Текстовый документ может иметь иерархическую структуру. Заголовки разделов документа служат узлами иерархического дерева. На нижних уровнях

иерархии находятся абзацы, таблицы и рисунки документа.

Структура позволяет быстро перемещаться по документу, автоматически формировать оглавление и редактировать состав документа.

Процесс создания структуры называется разметкой документа. Разметку рекомендуется проводить для больших документов. В процессе разметки обычным абзацам назначаются соответствующие уровни иерархии стили заголовков. Например, если назначить абзацу стиль **Заголовок 1**, то данный абзац будет находиться на первом уровне иерархии. Если назначить абзацу **Заголовок 2**, то данный абзац будет находиться на втором уровне иерархии и т.д.

Оглавление документа оформляется после создания структуры. В диалоговом окне **Оглавление и указатели** можно выбрать стиль оформления оглавления и количество уровней иерархии, которые будут отражаться в оглавлении.

2. Табличный процессор OpenOffice.org Calc.

Обработка данных средствами электронных таблиц.

Электронная таблица – это программа для математической, статистической и графической обработки текстовых и числовых данных.

Calc – это модуль электронных таблиц OpenOffice.org (OOo).

Документ OpenOffice.org Calc называется *рабочей книгой*. Рабочая книга представляет собой набор *рабочих листов*, каждый из которых имеет табличную структуру и может содержать одну или несколько таблиц. Каждый рабочий лист имеет свое *название* (на ярлыке листа). Чтобы переименовать рабочий лист, надо дважды щелкнуть на его ярлычке.

Рабочий лист состоит из **строк** и **столбцов**. Столбцы озаглавлены латинскими буквами (двухбуквенными комбинациями). Строки последовательно нумеруются цифрами. На пересечении столбцов и строк образуются *ячейки* таблицы – *минимальный элемент для хранения данных*. Обозначение ячейки – имя столбца и номер строки – выполняет функции *адреса ячейки* (например, A1; B5).

Одна из ячеек всегда является **активной** и выделяется рамкой – **табличным курсором**.

На рис. 7 представлено рабочее окно программы **Calc**, типичного приложения **OpenOffice.org**. Элементы пользовательского интерфейса – строка программного меню, панели инструментов, строка состояния – называются рабочей областью программы.

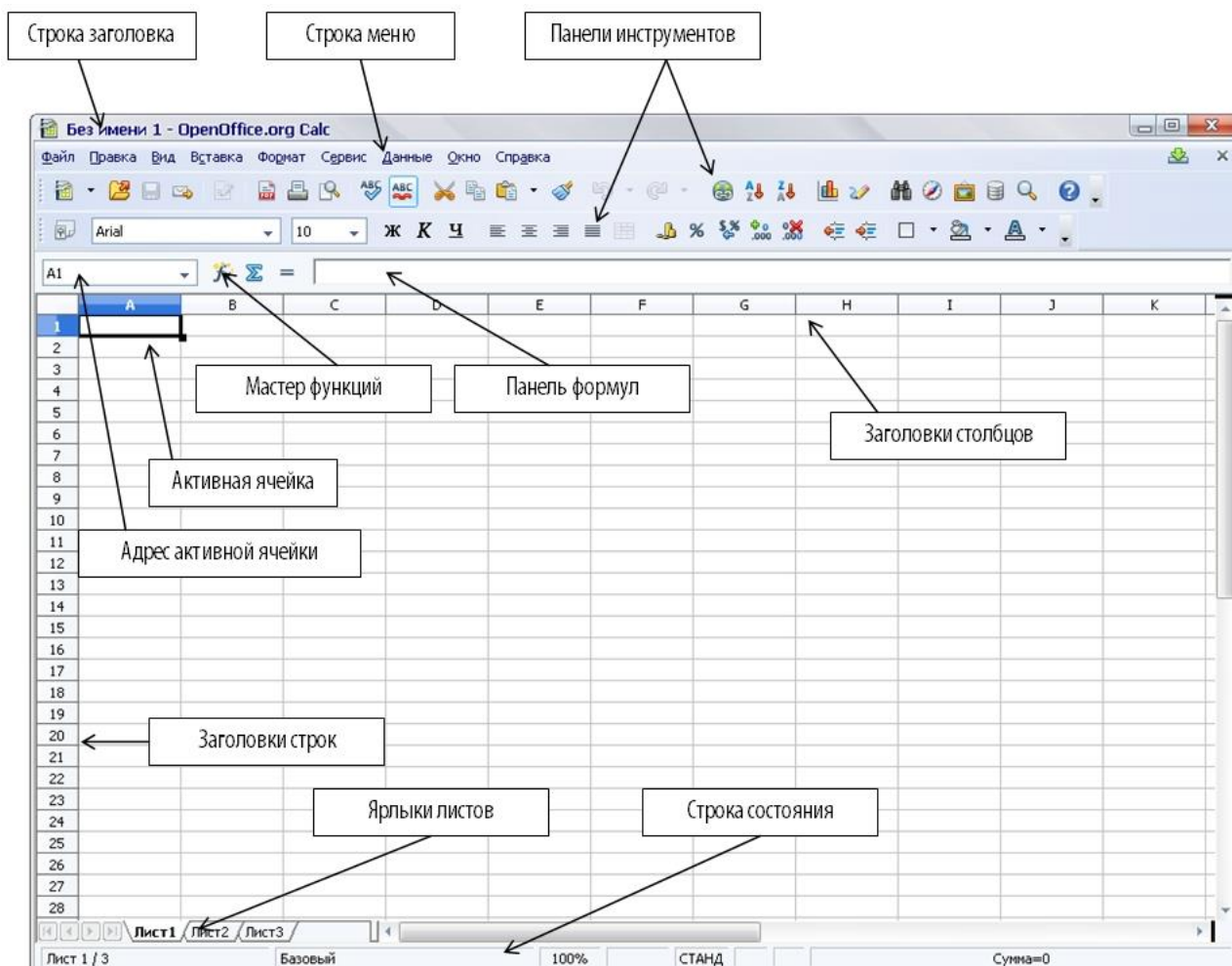


Рис. 7. Рабочее окно программы Calc

Панель формул – это специализированная панель, характерная именно для программы Calc. Она предназначена для ввода и изменения адресов ячеек, а также для работы с формулами.

Состав **панели Формул**:





С левой стороны Панели формул расположено небольшое текстовое поле, называемое *Область листа*, в котором находится адрес активной (текущей) ячейки.

Вторая часть представлена кнопками: **Мастер функций**, **Сумма**, **Функция**.

Нажатие на кнопку **Мастер функций** открывает диалоговое окно, в котором можно найти список возможных функций.

Кнопка **Сумма** вставляет в текущую ячейку сумму чисел, расположенных в верхних ячейках или в левых ячейках (если в верхних ячейках числа отсутствуют).

Кнопка **Функция** вставляет знак равенства в выбранную ячейку и **Строку ввода**, что определяет готовность ячейки к вводу формулы.

При вводе новых данных в ячейку, которая уже содержит данные, кнопки **Сумма** и **Функция** заменяются кнопками **Отменить** и **Принять**  

Содержимое текущей ячейки (данные, формула или функция) отображается в **Строке ввода**, которая является остальной частью *Панели формул*. Здесь можно редактировать содержимое текущей ячейки, что можно также делать и в самой текущей ячейке. Чтобы редактировать в области Строки ввода, нужно щелкнуть левой кнопкой по соответствующей части области Строки ввода, а затем ввести свои изменения. Чтобы редактировать в текущей ячейке, нужно просто щелкнуть дважды по этой ячейке.

Табличное поле

Центр окна программы Calc занимает рабочая область в виде табличного поля, разбитого на ячейки. В ячейках и происходит непосредственный набор текста, ввод данных, создание формул и функций, а также размещение рисунков, диаграмм и других объектов.

Ячейка – это наименьшая структурная единица электронной таблицы; она имеет адрес определяемый именем столбца и номером строки.

В ячейки таблицы вводятся **три типа данных**: *текст*, *число* и *формула*.

	А	В	С	Д
1	Наименование	Количество	Цена, руб.	Стоимость, руб.
2	Диск TDK	10	20	=B2*C2
3	Диск Verbatim	25	12,4	=B3*C3
4	Диск VS	100	9,8	=B4*C4

текст

число

формула

Ввод текста. По умолчанию текст выравнивается по левому краю ячейки.

Ввод чисел. Числа вводятся с помощью верхнего ряда клавиатуры или числовой клавиатуры. Для ввода отрицательного числа используется дефис. В качестве десятичного разделителя применяется запятая. Если перед числом ввести 0, то Calc его удалит. По умолчанию числа выравниваются по правому краю ячейки. Если после ввода число не помещается в ячейку, то в ней отображаются символы #####. Чтобы число вновь имело полный вид, необходимо увеличить ширину столбца.

Ввод формулы всегда начинается со знака =. Формула – это арифметическое или логическое выражение, содержащее константы, операторы, ссылки, функции, имена массивов.

Константа – постоянное (не вычисляемое) значение. Может быть числом или текстом.

Операторы – это знак или символ, определяющий тип вычисления в формуле над операндами.

Операнд – это элемент вычисления (константы, функции или ссылки).

Ссылка – адрес ячейки или диапазона ячеек на табличном поле, в которых содержатся значения.

Функция – заранее созданная формула, выполняющая сложные вычисления по введенным значениям (аргументам) в строго определенном порядке. Функции бывают математические, статистические, финансовые, логические и др.

Ссылки в формулах.

При создании связей между ячейками используется три типа ссылок: *относительные, абсолютные* и *смешанные*.

Относительная ссылка – это ссылка в формуле, основанная на относительном расположении ячейки, в которой находится формула, и ячейки, на которую указывает ссылка. При изменении позиции ячейки с формулой соответственно изменяется и ссылка на связанную ячейку. Так что, *например*, при копировании формулы вдоль столбцов или строк ссылка автоматически корректируется с учетом перемещения ячейки с формулой. Данный тип ссылок используется по умолчанию.

Абсолютная ссылка – это неизменная ссылка в формуле на ячейку, расположенную в определенном месте. При копировании (перемещении) ячейки с формулой адрес ячейки с абсолютной ссылкой не корректируется. Абсолютная ссылка указывается символом \$.

Для быстрого изменения типа адресации в формулах используется комбинация клавиш **Shift+F4**.

Например, абсолютная ссылка на ячейку \$A\$1 указывает на неизменность адреса ячейки A1 при копировании формулы вдоль столбца или строки.

Смешанная ссылка – это ссылка с использованием либо абсолютной ссылки на столбец и относительной – на строку (\$A1), либо абсолютной ссылки на строку и относительной – на столбец (A\$1). При изменении позиции ячейки с формулой относительная ссылка строки или столбца изменяется, а абсолютная часть остается прежней.

Относительная ссылка

	A	B	C	D		A	B	C	D
1	Наименование	Количество	Цена, руб.	Стоимость, руб.	1	Наименование	Количество	Цена, руб.	Стоимость, руб.
2	Диск TDK	10	20	=B2*C2	2	Диск TDK	10	20	200
3	Диск Verbatim	25	12,4	=B3*C3	3	Диск Verbatim	25	12,4	310
4	Диск VS	100	9,8	=B4*C4	4	Диск VS	100	9,8	980

Адреса ячеек при копировании меняются

Пример использования ссылок

Абсолютная ссылка

	A	B	C	D		A	B	C	D
1	price.ru		Курс \$	32,8	3	Наименование	Количество	Цена, руб.	Цена, \$
2					4	Диск TDK	10	200	6,10
3	Наименование	Количество	Цена, руб.	Цена, \$	5	Диск Verbatim	25	310	9,45
4	Диск TDK	10	200	=C4/\$D\$1	6	Диск VS	100	980	29,88
5	Диск Verbatim	25	310	=C5/\$D\$1					
6	Диск VS	100	980	=C6/\$D\$1					

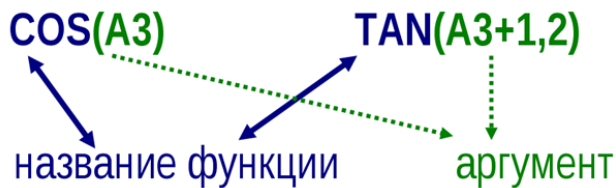
Адрес ячейки \$D\$1 при копировании не меняется

Работа с функциями

Функции – это заранее созданные формулы, которые выполняют вычисления по введенным величинам и в указанном порядке.

Каждая функция состоит из следующих элементов:

- названия функции;
- аргументов.



Каждая функция имеет свой синтаксис и порядок действия, который нужно соблюдать, иначе вычисления будут ошибочными. Имя функции можно вводить вручную с клавиатуры, а можно выбирать в окне **Мастер функций**.

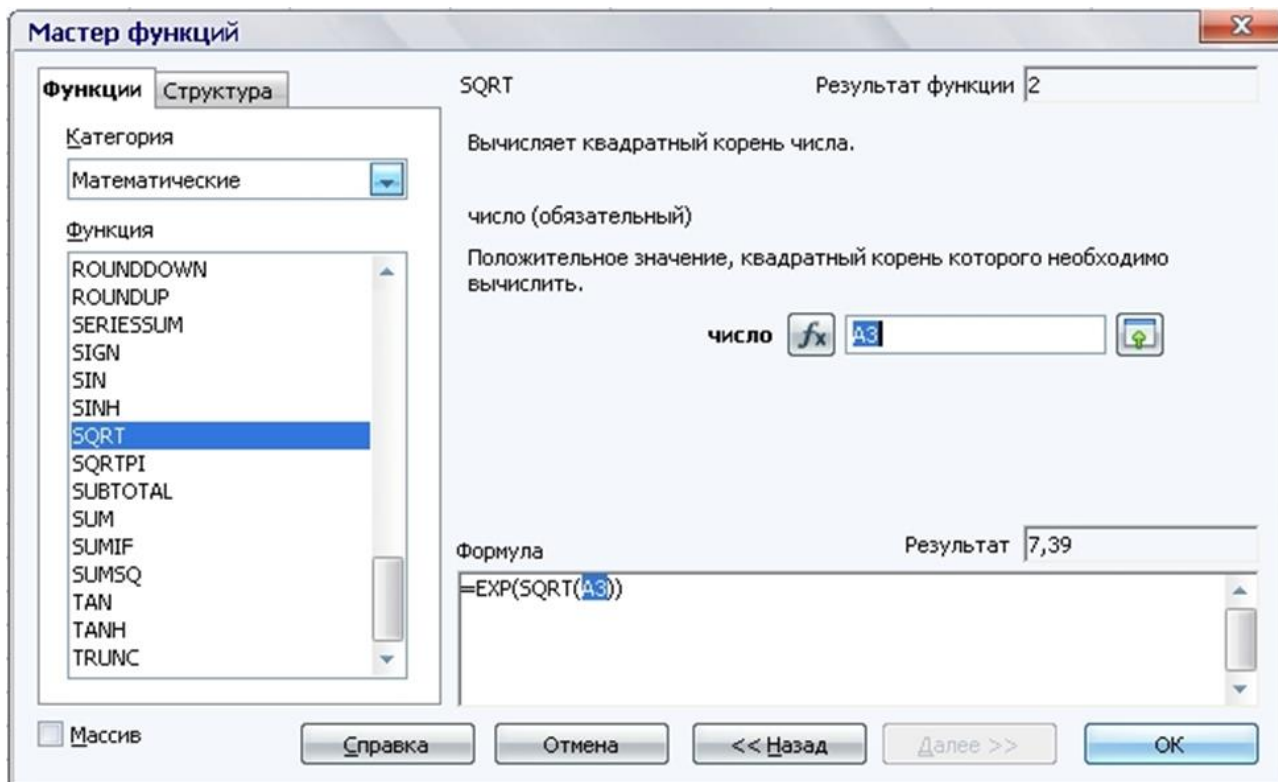


Рис. 8. Диалоговое окно Мастер функций

Функция может выступать в качестве аргумента для другой функции. В этом случае она называется вложенной функцией (см. рис. 8). При этом в формулах можно использовать до нескольких уровней вложенной функции.

При работе с вложенными функциями появляется всплывающая подсказка аргумента, щелкнув по которой можно открыть окно справки, помогающей разобраться с синтаксисом и аргументами.

Ошибочные значения

Сообщение об ошибке	Ошибка
#DIV/0!	Деление числа на ноль. =5/0
#ИМЯ?	Не может распознать имя, используемое в формуле. =Я1*4
#ЗНАЧЕН!	Введена математическая формула, которая ссылается на текстовое значение.
####	Ширины ячейки не хватает для отображения содержимого.
Ошибка 501	Недопустимый символ в формуле.
Ошибка 502	Функция имеет недопустимый аргумент, например отрицательное число для функции извлечения корня.
Справка – Коды ошибок в OpenOffice.org Calc	

3. OpenOffice.org Impress.

Программные технологии создания мультимедиа-презентаций.

Приложение OpenOffice.org Impress позволяет создавать слайд-шоу с использованием диаграмм, рисованных объектов, анимации, текста, мультимедиа и множества других элементов.

На рис. 9 представлено рабочее окно программы.

Панель слайдов – область для размещения страниц с добавленными в презентацию слайдами. Ее можно использовать для добавления, переименования, удаления и перегруппировки слайдов или страниц. Для активации панели слайдов используется команда в меню **Вид – Панель слайдов**.

Центр окна программы Impress занимает **рабочая область** в виде страницы слайда, имеющая пять режимов отображения:

- **Режим рисования** – основной режим приложения для создания слайда. Внизу отображается панель **Рисование**.
- **Режим структуры** – режим редактирования структуры презентации. Он включается с панелью **Структура**, которая содержит часто используемые инструменты редактирования: масштаб, уровни и т.д. В данном режиме удобно задавать и редактировать заголовки слайдов.

- **Режим примечаний** – режим внесения служебных примечаний, недоступных при демонстрации презентации.
- **Режим тезисов** – режим для подготовки слайдов к печати на принтере.
- **Сортировщик слайдов** – режим перегруппировки слайдов для подготовки к демонстрации.

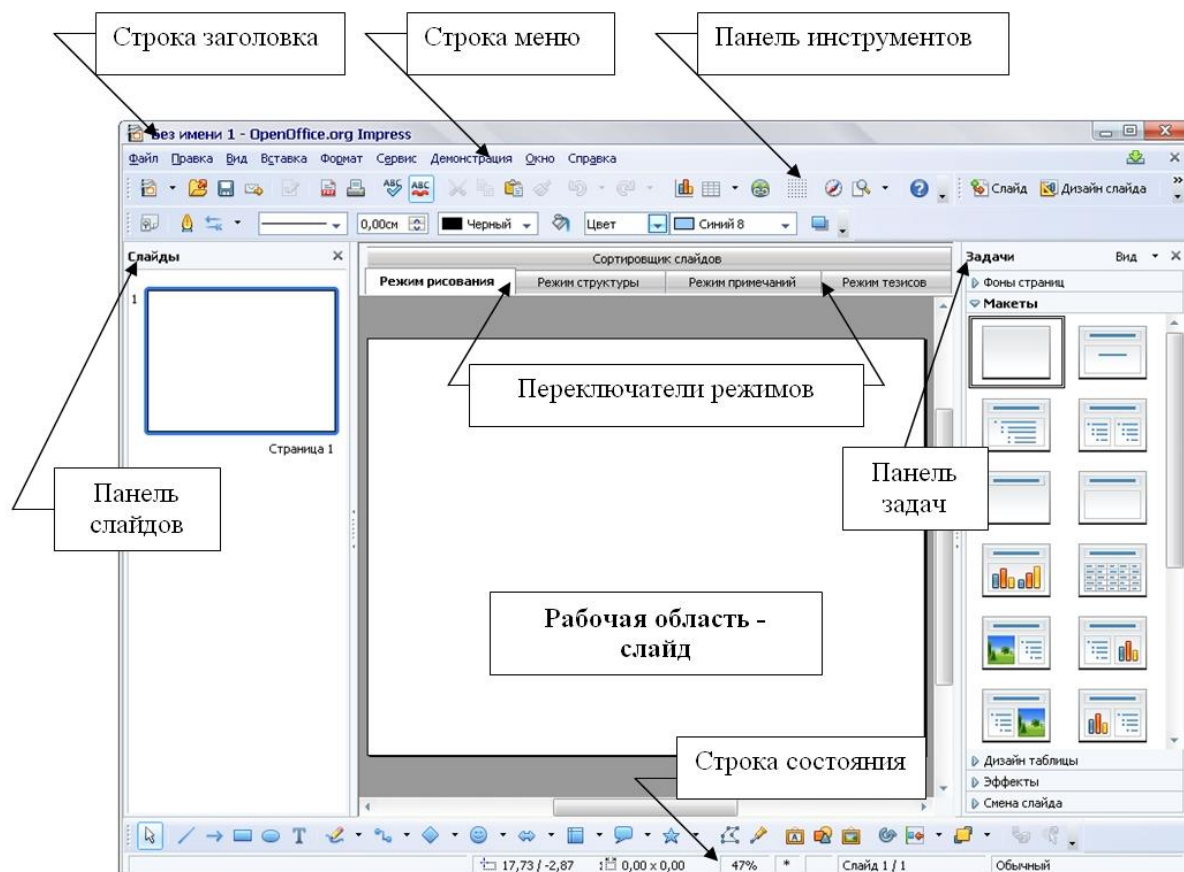


Рис. 9. Рабочее окно программы Impress

Панель задач – область для создания и редактирования слайдов. В ней содержатся страницы-шаблоны, служащие для создания слайда, его разметки, переходов между слайдами и анимационных эффектов для объектов на слайдах.

Панель задач состоит из пяти разделов: **Фона страниц**, **Макеты**, **Дизайн таблицы**, **Эффекты** и **Смена слайда**.

Для активации панели задач используется команда в меню **Вид – Панель задач**.

Строка состояния – это служебная панель, которая располагается внизу окна приложения. Строка состояния отображает информацию о презентации, включая выделенный в данный момент объект.

Новую презентацию можно создавать при помощи мастера или же стандартным путем.

Мастер служит для пошагового создания презентаций, помогая применять шаблоны презентаций, разрабатывать макет и использовать различные возможности для редактирования.

На каждой странице мастера можно отменять, изменять и пропускать любые операции редактирования. В случае пропуска страниц мастер будет использовать параметры по умолчанию.

В результате запуска приложения OpenOffice.org Impress открывается окно **Мастер презентаций 1**. (рис. 10) в группе **Тип** выбрать тип презентации:

- **Пустая презентация** – для создания новой презентации;
- **Из шаблона** – для создания новой презентации на основе существующих шаблонов. Список шаблонов становится доступным в дополнительном поле;
- **Открыть существующую презентацию** – для открытия существующей презентации. Поиск презентаций начинается после щелчка по кнопке **Открыть** в окне **Открыть**.

Для просмотров типов презентаций рекомендуется активировать строку **Предварительный просмотр**.

После выбора типа презентации щелкнуть по кнопке **Далее**.

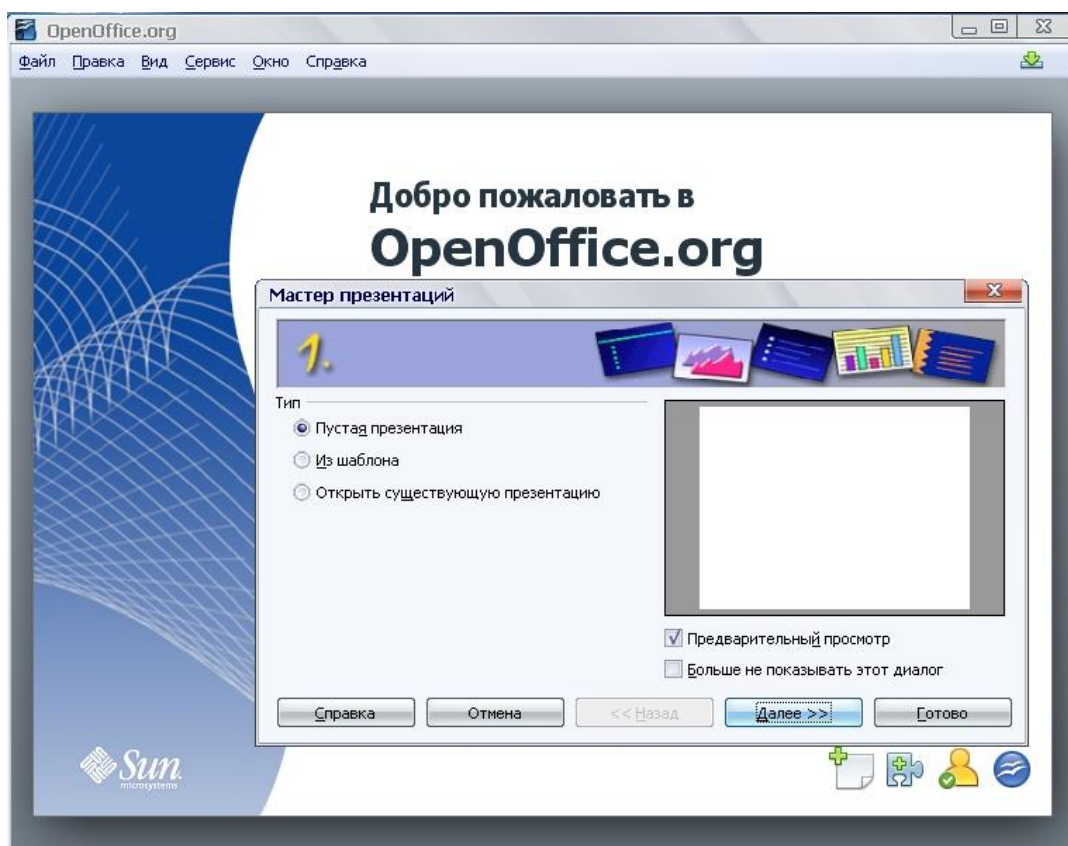


Рис. 10. Окно Мастер презентаций 1.

В окне **Мастер презентаций 2.** (рис. 11) в группе **Выберите стиль слайда** сначала в верхнем списке выбрать один из двух типов дизайна: *презентацию* или *фон презентации*. Затем в нижнем списке щелкнуть по нужной строке списка стилей дизайна для презентации.

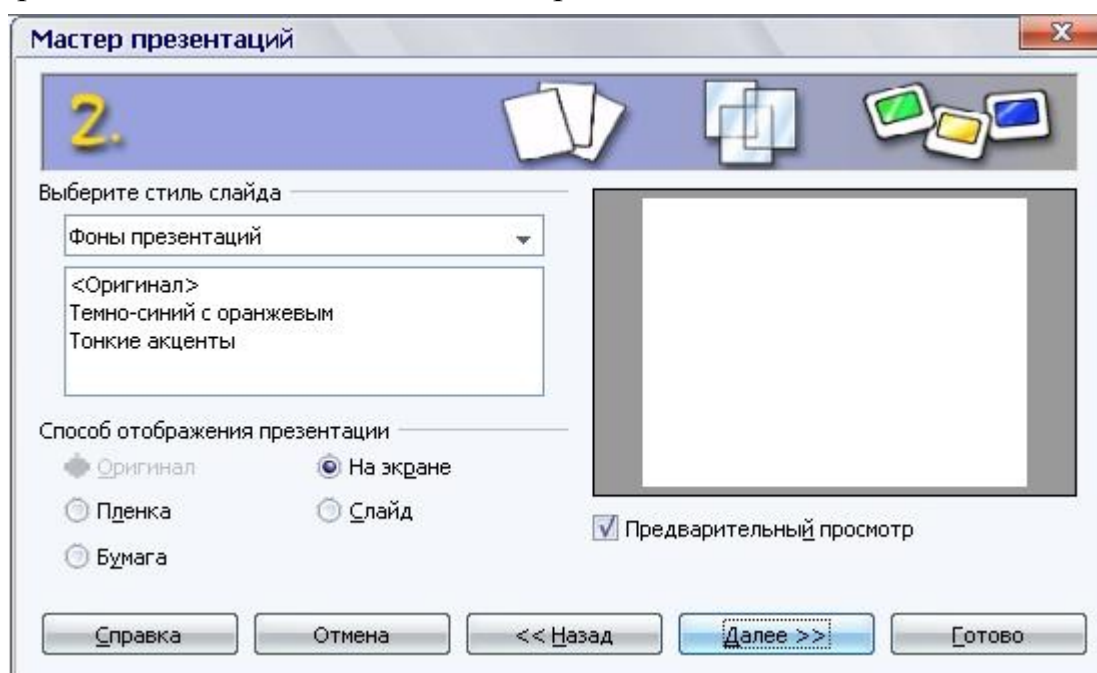


Рис. 11. Окно Мастер презентаций 2.

В группе **Способ отображения презентации** выбрать:

- **Оригинал** – для использования исходного формата страниц для шаблона;
- **Пленка** – для создания презентации с применением прозрачных пленок;
- **Бумага** – для создания презентации в печатном виде;
- **На экране** – для создания презентации только на экране компьютера;
- **Слайд** – для создания презентации в виде слайдов.

После выбора стиля презентации и ее носителя щелкнуть по кнопке **Далее**.

В окне **Мастер презентаций 3.** (рис. 12) в группе **Выберите тип смены слайда** в списке **Эффект** выбрать специальный эффект для презентации.

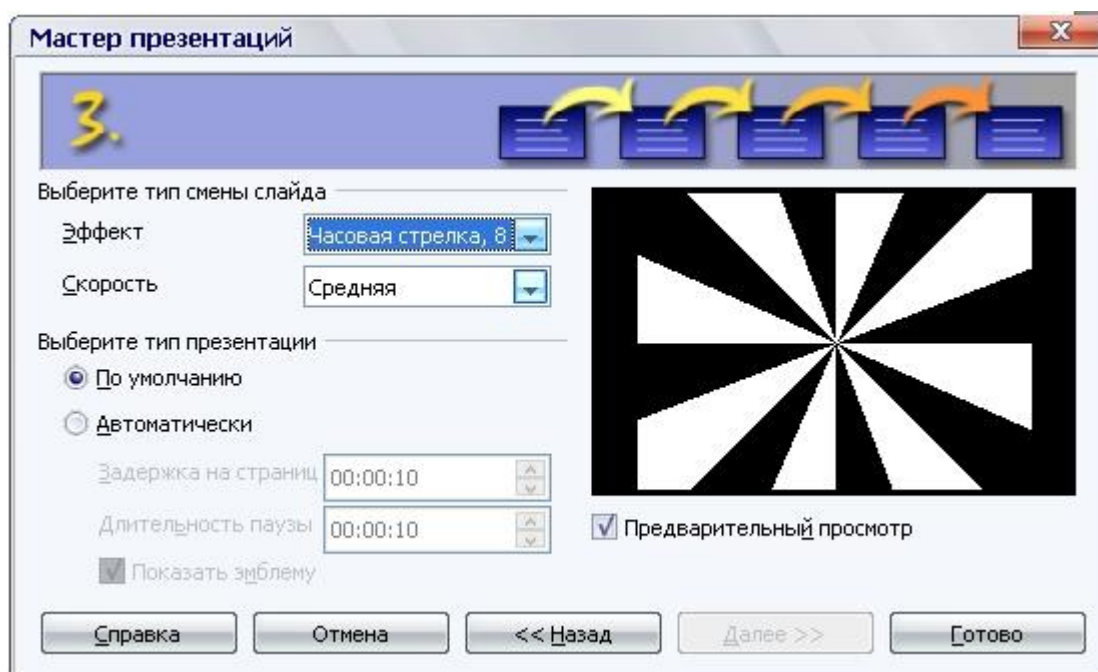


Рис. 12. Окно **Мастер презентаций 3.**

В списке **Скорость** определить скорость срабатывания эффекта для презентации.

В группе **Выберите тип презентации** определить время презентации:

- **По умолчанию** – для запуска презентации в полноэкранном режиме с указанной скоростью;
- **Автоматически** – для запуска презентации с перезапуском после перерыва.

Регулятором **Задержка на странице** задать длительность показа каждой страницы презентации, а регулятором **Длительность паузы** – длительность пауз между презентациями.

Если в окне **Мастер презентаций 1.** (рис. 10) был выбран пункт **Пустая презентация**, то действие мастера на этом этапе заканчивается, и необходимо щелкнуть по кнопке **Готово**, а если были выбраны другие типы – по кнопке **Далее**.

В окне **Мастер презентаций 4.** (рис. 13) описать основные идеи презентации.

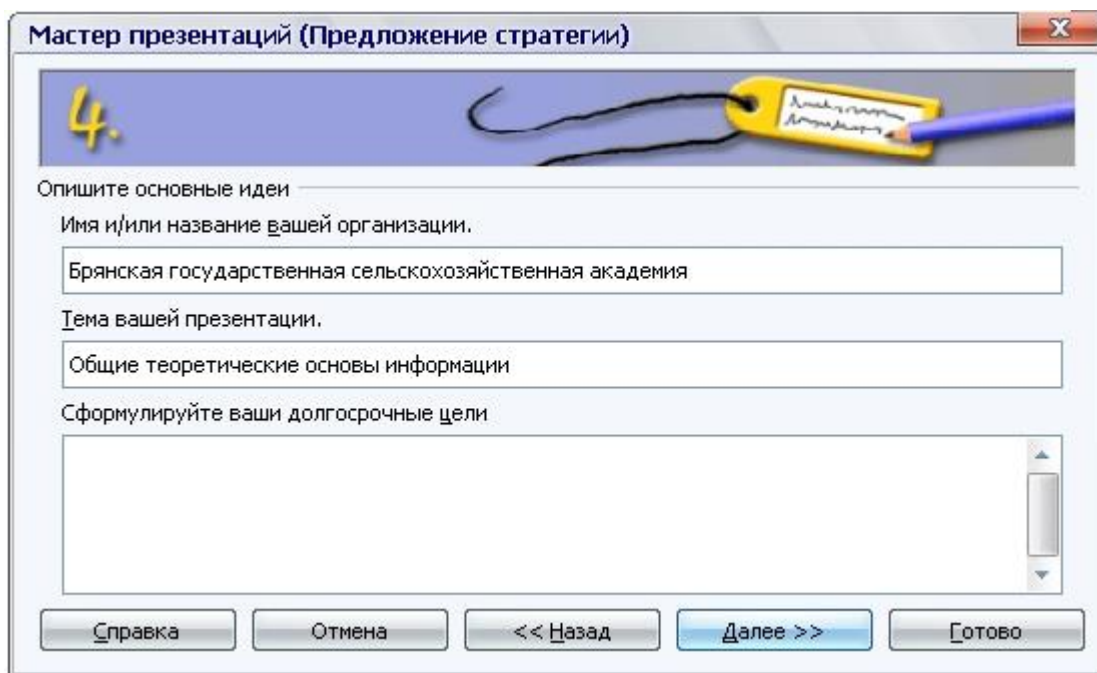


Рис. 13. Окно Мастер презентаций 4.

В окне **Мастер презентаций 5.** (рис. 14) в группе **Выберите страницы** отобразится список всех страниц, принадлежащих выбранному шаблону.

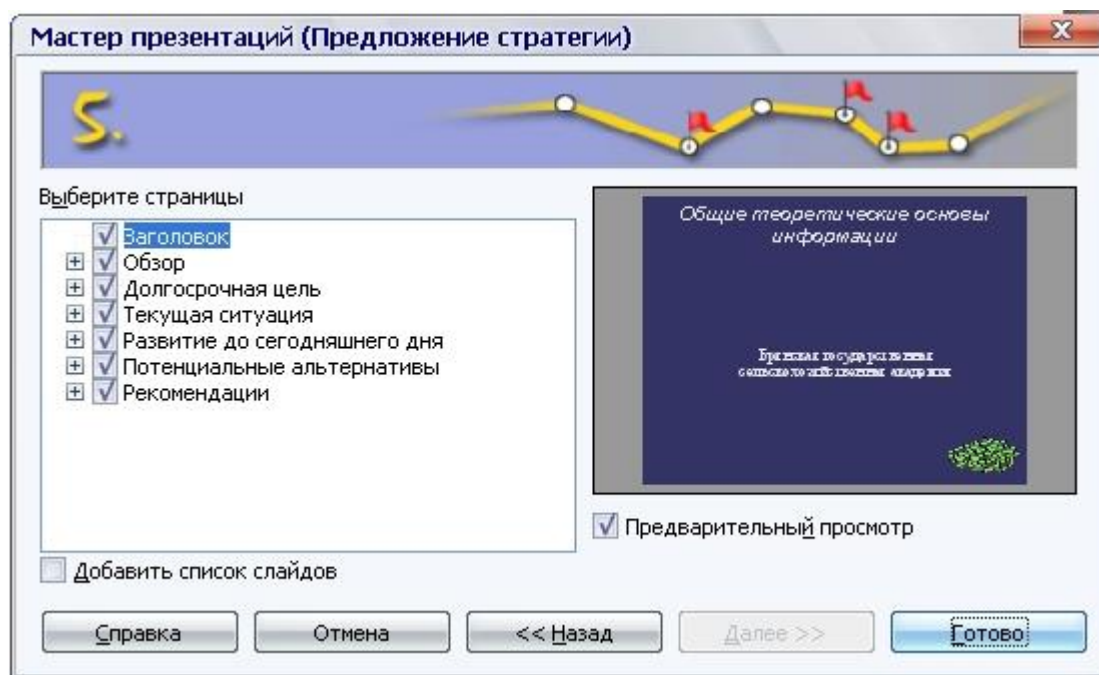


Рис. 14. Окно Мастер презентаций 5.

Для исключения страницы из презентации отключить флажок перед ее именем. Для просмотра страницы следует щелкнуть по маленькому плюсу рядом с именем страницы.

Щелкнуть по кнопке Готово.

По умолчанию документ сохраняется в формате **.odp** (формат OpenOffice.org Impress). Документ, созданный в формате OpenOffice.org, можно сохранить в формате Microsoft Office, используя команду **Файл – Сохранить как**. В открывшемся диалоговом окне в списке **Тип файла** выбрать строку Microsoft PowerPoint 97/2000/XP (*.ppt). Указать имя файла и место сохранения его и щелкнуть по кнопке Сохранить.

Оформление слайда

Каждому слайду презентации можно задать **макет**, который определяет стиль форматирования текста для названия и структуры, их взаимное расположение.

В окне открытой презентации в **Режиме рисования** перейти к нужной странице и на панели задач раскрыть область **Макеты**. В палитре макетов (рис. 15) выбрать нужный образец.

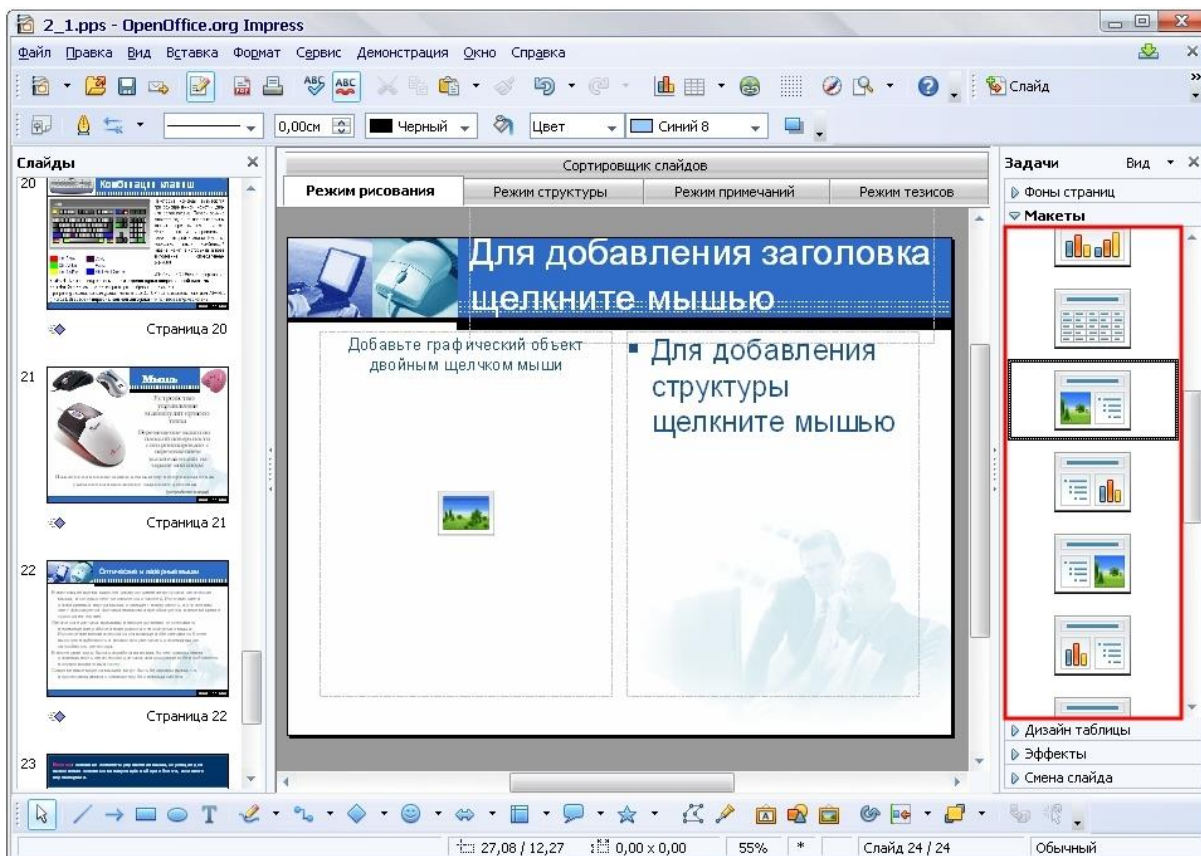


Рис. 15. Окно Impress. Палитра макетов.

Применение эффектов к используемым в презентации объектам.

Добавление эффекта к объекту.

1. В окне открытой презентации в режиме рисования выделить нужный объект.
2. На панели задач открыть область **Эффекты** и щелкнуть по кнопке **Добавить**.
3. В окне **Эффекты** (рис. 16) задать на вкладках:
 - Вступление – эффект первого появления объекта;
 - Выделение – эффект выделения объекта;
 - Выход – эффект выхода объекта;
 - Пути движения – эффект перемещения объекта;
 - Дополнительные эффекты – предварительный просмотр новых или измененных эффектов.
4. На каждой вкладке при необходимости задать скорость срабатывания эффекта. Закрыть окно кнопкой **Ок**.

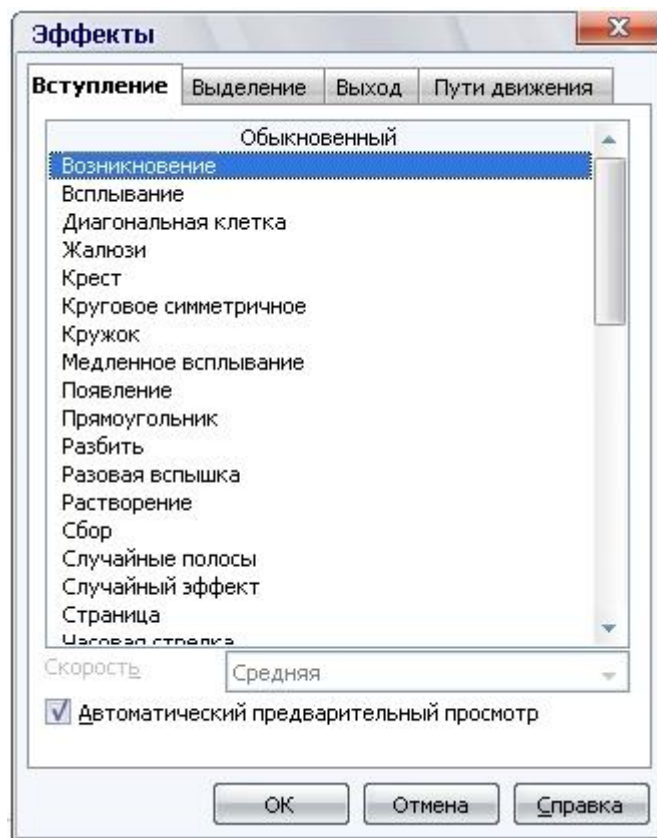


Рис. 16. Окно Эффекты

5. В области **Эффекты** на панели задач, где отобразится выбранный эффект, открыть список графы **Начать** и выбрать способ запуска эффекта:
 - При щелчке – для запуска эффекта щелчком мыши;
 - Вместе с предыдущим – для запуска эффекта сразу;
 - После предыдущего – для запуска эффекта после окончания предыдущего.

6. Для эффектов перемещения в графе **Направление** выбрать направление движения объекта.

Просмотр эффекта.

1. В окне открытой презентации в режиме рисования выделить нужный объект с заданным эффектом.
2. На панели задач открыть область **Эффекты** и щелкнуть по кнопке **Воспроизвести**.

Изменение эффекта объекта.

1. В окне открытой презентации в режиме рисования выделить нужный объект с эффектом.
2. На панели задач в области **Эффекты** щелкнуть по кнопке **Изменить**.

3. В окне **Эффекты** задать на вкладках новые параметры. Закрыть окно щелчком по кнопке **Ок**.

Изменение направления эффекта и скорость его исполнения можно поменять в соответствующих списках области **Эффекты** на панели задач.

Применение эффектов к слайдам.

1. В окне открытой презентации выбрать страницу нужного слайда на панели слайда.
2. Открыть область **Смена слайда** на панели задач.
3. В списке эффектов выбрать строку нужного эффекта. Рекомендуется активировать пункт **Автоматический просмотр** для немедленного просмотра выбранного эффекта.
4. Если необходимо использовать выбранный эффект для всех слайдов презентации, следует щелкнуть по кнопке **Применить ко всем слайдам**.
5. В группе **Изменить переход** в списке **Скорость** выбрать имя срабатывания эффекта.
6. В списке **Звук** при необходимости выбрать звуковое сопровождение для эффекта.
7. В группе **Смена слайдов** выбрать способ смены слайдов при демонстрации:
 - По щелчку – для перехода к следующему слайду щелчком мыши;
 - Автоматически – для перехода к следующему слайду после заданной регулятором паузы.

Для предварительного просмотра смены слайдов необходимо щелкнуть по кнопке **Воспроизвести**.

Демонстрация презентации – это последовательный настраиваемый показ слайдов с использованием эффектов.

Задание параметров демонстрации слайдов.

1. В окне открытой презентации открыть меню **Демонстрация – Параметры демонстрации**.
2. В окне **Настройки демонстрации** (рис. 17) при необходимости активировать:
 - Все слайды – для показа всех слайдов презентации;
 - Начать со слайда – для демонстрации презентации с определенного слайда, заданного в списке справа от данного пункта.

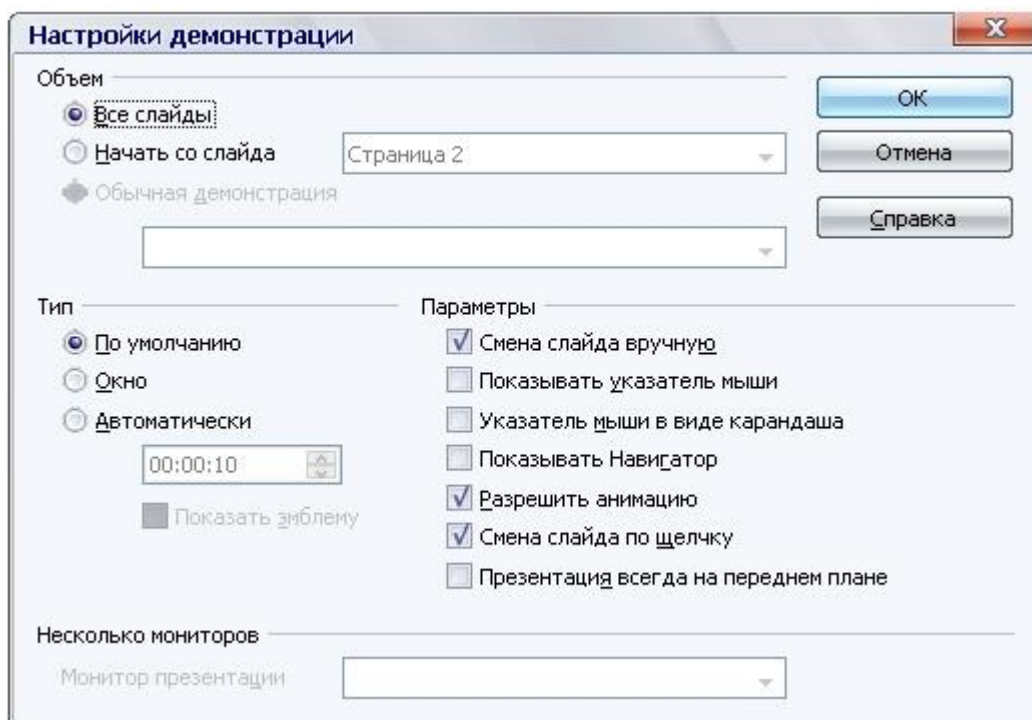


Рис. 17. Окно Настройки демонстрации

3. В группе **Тип** выбрать тип показа слайдов:
 - По умолчанию – для отображения слайдов во весь экран;
 - Окно – для отображения слайдов в окне программы OpenOffice.org;
 - Автоматически – для повторной демонстрации слайдов после заданной паузы.
4. Для отображения на слайде паузы эмблемы OpenOffice.org активировать пункт **Показать эмблему**.
5. В группе **Параметры** при необходимости установить:
 - Смена слайдов вручную – для смены при демонстрации слайдов вручную;
 - Показывать указатель мыши – для отображения указателя мыши во время показа слайдов;
 - Указатель мыши в виде карандаша – для преобразования указателя мыши в перо, которым можно рисовать на слайдах во время презентации;
 - Показывать Навигатор – для отображения окна Навигатор во время показа слайдов;

- Разрешить анимацию – для отображения анимированных кадров во время показа слайдов;
- Смена слайда по щелчку – для перехода к следующему слайду щелчком мыши;
- Презентация всегда на переднем плане – для расположения окна OpenOffice.org во время презентации на переднем плане.

В списке **Несколько мониторов** при необходимости выбрать монитор для использования в полноэкранном режиме демонстрации. Закрывать окно щелчком по кнопке **Ок**.

Автоматическая демонстрация слайдов.

Автоматический переход от слайда к слайду следует настроить для каждого слайда.

1. В окне открытой презентации на панели задач раскрыть область **Смена слайда**, выбрать пункт **Автоматически через**, а затем регулятором задать нужное время начала демонстрации презентации.
2. Щелкнуть по кнопке **Применить ко всем слайдам**.
3. Каждому слайду можно задать собственное время перехода.

Запуск демонстрации презентации.

1. В окне открытой презентации раскрыть меню **Демонстрация**.
2. В списке команд выбрать пункт **Демонстрация** или можно использовать клавишу **F5**.
3. Для приостановки демонстрации следует использовать клавишу **Esc**.

Контрольные вопросы

1. Каковы основные функции текстового процессора?
2. В каком формате нужно сохранить файл, чтобы он мог быть прочитан в других приложениях с сохранением форматирования? Без сохранения форматирования?
3. Возможно ли создать текстовый документ, состоящий из страниц различного формата?
4. В каком случае изменится иерархическая структура оглавления документа?
5. Каким требованиям должна отвечать современная электронная таблица?
6. Назовите способы автоматизации ввода данных.
7. Какие существуют форматы ячеек?
8. Какова основная и дополнительная функциональность программы для создания и демонстрации презентаций?
9. Какие существуют режимы отображения слайдов и для чего они предназначены?
10. Укажите способы создания презентаций? Как используется Мастер презентаций?

Лекция 7.

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

1. Основы алгоритмизации процессов обработки информации.
2. Основные понятия и классификация языков программирования.
3. Основные объекты языка программирования QBASIC.
4. Примеры программирования на языке QBASIC.

1. Основы алгоритмизации процессов обработки информации.

Один из важнейших этапов решения задачи на компьютере заключается в разработке алгоритма.

Понятие *алгоритма* является одним из основных в математике, оно возникло задолго до появления ЭВМ. В настоящее время оно представляет собой одну из основных концепций информатики и приобрело широкую известность в связи с распространением современной компьютерной техники и связанных с ней технологий обработки информации.

Алгоритм – это четкое описание строго определенной последовательности действий, которые необходимо выполнить для преобразования исходных данных в результат решения задачи.

Для алгоритма характерны следующие основные *свойства*:

Дискретность. Это свойство означает, что при составлении алгоритма процесс решения задачи представляется в виде последовательности шагов (этапов). Для выполнения каждого шага требуется определенный отрезок времени.

Определенность (детерминированность), т.е. точность и однозначность. Это свойство состоит в том, что каждый шаг алгоритма должен быть определен точно и однозначно и не оставлять места для произвольного толкования его исполнителем. В связи с этим процесс выполнения алгоритма носит механический характер.

Результативность (конечность). Это свойство означает, что алгоритм должен приводить к цели за конечное число достаточно простых шагов, понятных исполнителю. Часто это свойство усиливается, и говорят, что алгоритм должен быть выполнен не просто за конечное, а за разумное время.

Массовость. Это свойство отражает тот факт, что алгоритм должен быть применим для решения любой задачи из некоторого класса, т.е. алгоритм должен разрабатываться в достаточно общем виде для решения целого ряда задач, различающихся исходными данными.

При решении большинства задач не удастся одновременно разрабатывать и описывать алгоритм на языке программирования (алгоритмическом языке), понятном ЭВМ. Поэтому алгоритм предварительно должен быть записан одним из способов, облегчающих его перевод на машинный язык.

Традиционно сложились следующие **основные способы записи алгоритмов:**

- *на естественном языке,*
- *на формальном языке (язык программирования),*
- *на графическом языке (блок-схемы).*

Наиболее широко используется графическое описание алгоритма в виде блок-схемы, при котором этапы процесса обработки информации изображаются в виде геометрических фигур (символов, или, как их еще называют, блоков или блочных символов). Связи между этапами и последовательность их выполнения указываются линиями, соединяющими блоки. Блочный символ имеет различные графические изображения в зависимости от выполняемых действий. Внутри символов словами или с помощью формул записываются действия, составляющие содержание этапов алгоритма. Графическое изображение символов, применяемых при составлении блок-схемы, установлено ГОСТом 19.002 – 80 и 19.005 – 80. Основные графические символы, используемые для описания алгоритмов, приведены в таблице 1.

Основными направлениями выполнения алгоритма считаются направления сверху вниз и слева направо. Если линии потока идут в этих направлениях и не имеют изломов, то они могут не заканчиваться стрелками. В остальных случаях применение стрелок обязательно.

Основное достоинство блок-схем в том, что с их помощью можно наглядно изобразить структуру алгоритма, т.е. сущность основных этапов вычислений и последовательность их выполнения. Степень детализации алгоритма при составлении блок-схемы зависит каждый раз от конкретных условий.

Таблица 1. ***Графическое изображение символов, используемых при составлении блок-схемы***

Операция	Графическое обозначение	Функция
Пуск-остановка		Начало, конец, прерывание процесса обработки данных или выполнения программы
Ввод-вывод		Преобразование данных в форму, пригодную для ввода или вывода
Процесс		Выполнение операции или группы операций
Решение		Выбор направления выполнения алгоритма в зависимости от некоторых условий
Модификация		Организация циклического процесса
Соединитель		Разрыв линий потока
Комментарий		Пояснения к функциональным блокам

Можно выделить два основных класса объектов, над которыми производятся действия при выполнении алгоритма: **константы**, которые не могут изменять своих значений в ходе выполнения алгоритма, и **переменные**, которые могут принимать различные значения при выполнении алгоритма. В соответствии с этим при записи алгоритма константы изображаются своими значениями, а переменные обозначаются именами.

Существуют три основных типа процессов обработки информации:

- **линейный**
- **разветвляющийся**
- **циклический**

Им соответствуют **базовые структуры (конструкции) алгоритмов**:

- **следование**
- **ветвление**
- **цикл**

Алгоритм любой степени сложности можно разработать, используя базовые конструкции.

Наиболее простым является *линейный процесс* обработки информации, при котором существует единственный путь выполнения, действия, производятся последовательно, одно за другим, и каждый этап алгоритма выполняется только один раз.

Алгоритм, называется **линейным**, если все его действия выполняются последовательно одно за другим.

Соответствующая линейному процессу блок-схема структуры следования представлена на рис. 1.

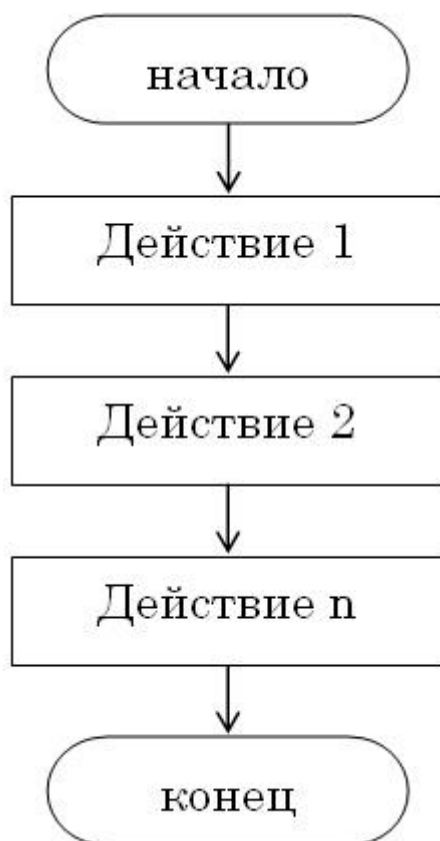


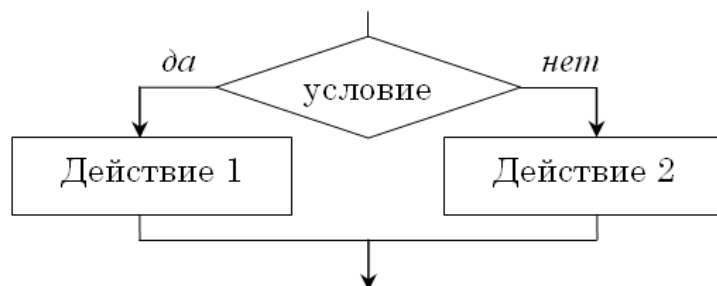
Рис. 1. Блок-схема структуры следования (линейный процесс обработки информации)

При составлении алгоритмов часто возникает необходимость направлять процесс обработки информации по одному из двух возможных путей, т.е. выполнить те или иные действия в зависимости от некоторого условия. Так как результат выполнения условия заранее неизвестен, то его проверка должна осуществляться в ходе алгоритма. Процессы такого типа называют *разветвляющимися*.

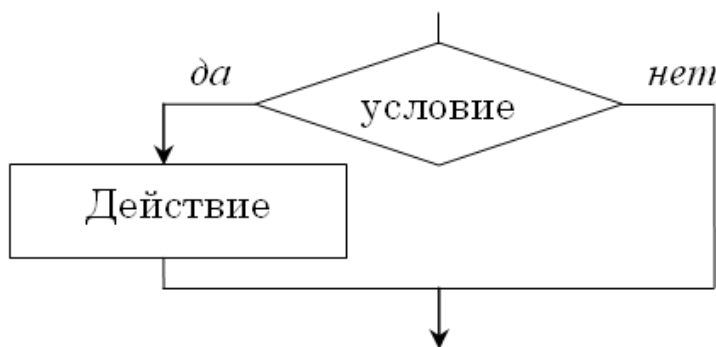
Алгоритм, называется **разветвляющимся**, если порядок выполнения шагов алгоритма изменяется в зависимости от заданных условий.

Блок схемы базовых структур ветвления представлены на рис. 2.

а)



б)



в)

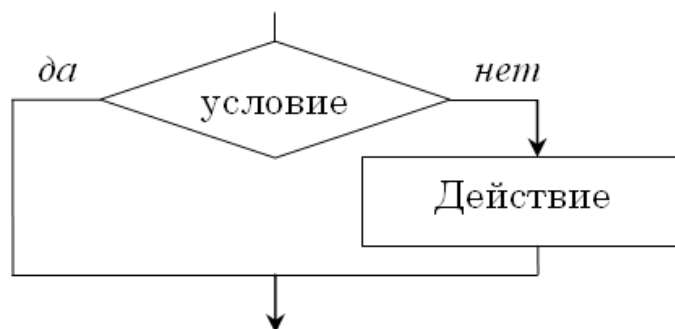


Рис. 2. Блок-схемы базовых структур ветвления:

а – общая структура ветвления; б, в – частные случаи ветвления

На рис. 2а представлена полная форма структуры ветвления, когда в каждой из ветвей содержится некоторое действие. Если условие истинно, то будет выполнено только действие 1, в противном случае будет выполнено только действие 2.

Если в ветвлении присутствуют команды только для случая истинности или только для случая ложности условия, то говорят, что она записана в сокращенной форме. На рис. 2б и 2в представлены частные случаи ветвлений, когда одна из ветвей «да» или «нет» не содержит никаких действий. Каждое из действий, выполняемое в той или иной ветви, может содержать несколько этапов алгоритма.

Алгоритмическая структура «*выбор*» применяется для реализации ветвления со многими вариантами групп действий. В структуру выбора входят несколько условий, проверка которых осуществляется в строгой последовательности их записи в команде выбора. При истинности одного из условий выполняется соответствующая последовательность действий.

В алгоритмической структуре «**выбор**» выполняется одно из нескольких последовательностей действий при истинности соответствующего **условия**.

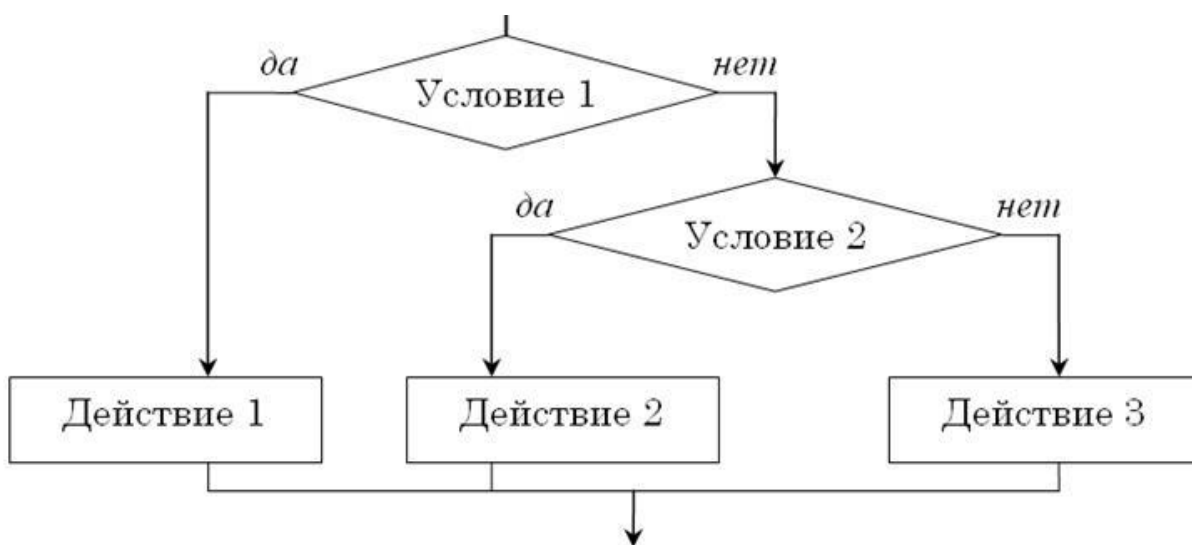


Рис. 3. Блок-схема алгоритмической структуры «выбор»

При составлении алгоритмов решения многих задач требуется многократно выполнить одни и те же действия по обработке информации. Процессы такого типа называют *циклическими*, а действие (группа действий), которое в процессе выполнения алгоритма непрерывно повторяется, называется *циклом*. **Цикл** представляет собой алгоритмическую конструкцию, в которой многократно выполняется одна и та же последовательность действий, называемая **телом цикла**.

Алгоритм, называется **циклическим**, если последовательность шагов алгоритма выполняется многократно.

Каждое однократное исполнение тела цикла называется *итерацией*.

Цикл с параметром. Когда заранее известно, какое число повторений тела цикла необходимо выполнить, можно воспользоваться циклической структурой представленной на рис. 4.

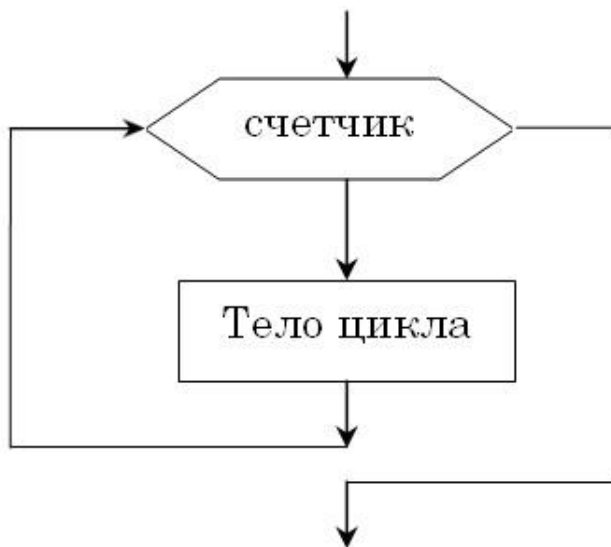


Рис. 4. Блок-схема цикла с параметром

В начале выполнения цикла значение переменной Счетчик устанавливается равным НачЗнач. При каждом проходе цикла переменная Счетчик увеличивается на величину шага. Если она достигает величины, большей КонЗнач, то цикл завершается и выполняются следующие за ним операторы.

Цикл с условием. Часто бывает так, что необходимо повторить тело цикла, но заранее неизвестно, какое количество раз это надо сделать. В таких случаях количество повторений зависит от некоторого условия. Условие выхода из цикла можно поставить в начале, перед телом цикла. Такой цикл называется *циклом с предусловием* (рис. 5).

Для того чтобы определить момент прекращения выполнения тела цикла, используется условие цикла. Если при истинности условия цикл продолжается, то такое условие называется *условием продолжения цикла*. Иными словами, такой цикл продолжается, пока условие цикла истинно.

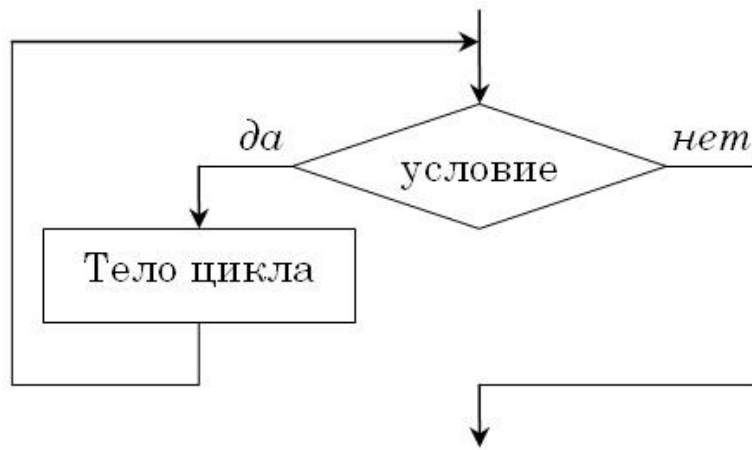


Рис. 5. Блок-схема цикла с предусловием продолжения

Если при истинности условия цикл завершается, то такое условие называется *условием завершения цикла*. В этом случае цикл продолжается до тех пор, пока условие цикла не станет истинным.

Условие выхода из цикла можно поставить в конце, после тела цикла. Такой цикл называется *циклом с постусловием* (рис. 6).

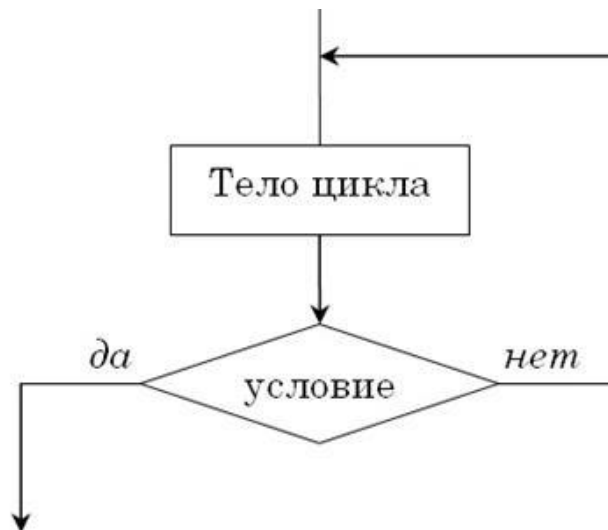


Рис. 6. Блок-схема цикла с постусловием завершения

Цикл с постусловием, в отличие от цикла с предусловием, выполняется обязательно как минимум один раз, независимо от того, выполняется условие или нет.

2. Основные понятия и классификация языков программирования.

Язык программирования – это формализованный язык, предназначенный для описания программ и алгоритмов решения задач на компьютере.

Компьютерная программа – это последовательность инструкций на машинно-ориентированном языке, предназначенных для выполнения центральным процессором вычислительной машины.

Программирование – это совокупность процессов, связанных с разработкой программ и их реализацией.

На рис. 1 показан приблизительный порядок действий при программировании. Программирование заключается в написании (исходного кода) программы на выбранном языке программирования.

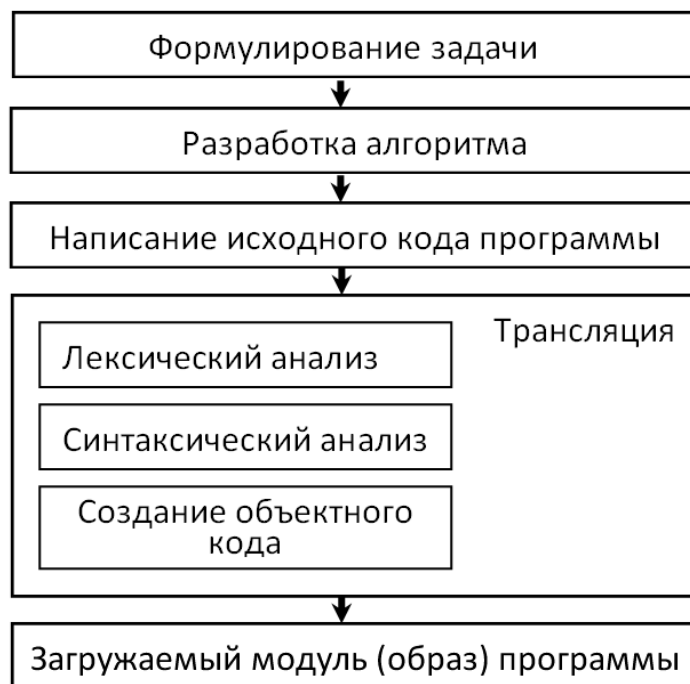


Рис. 7. Схематичное описание процесса программирования

Исходный код – текст компьютерной программы, написанный на выбранном языке программирования.

Исходный код преобразуется транслятором (специальной программой) в объектный модуль, который хранится на диске.

Транслятор – программа, осуществляющая трансляцию (перевод) компьютерной программы с языка программирования в программу на машинном коде. Трансляторы бывают двух видов – **компиляторы**¹ и **интерпретаторы**² (см. [лекцию 4](#). Классификация и тенденции развития программного обеспечения).

¹ **Компилятор** – осуществляет преобразование программы, написанной на языке программирования, в программу в машинных кодах.

² **Интерпретатор** – осуществляет непосредственное выполнение программы на языке программирования.

Для выполнения программы должен быть подготовлен объектный модуль – определены адрес его загрузки в оперативную память и адреса связи с другими объектными модулями. Затем загружается объектный модуль в оперативную память.

Объектный модуль – последовательность инструкций машинного кода, являющаяся результатом работы компилятора.

Машинный код – язык конкретной вычислительной машины (центрального процессора), инструкции которого выполняются непосредственно центральным процессором или его микропрограммами.

Исполняемый код – объединенные в загружаемую программу модули объектного кода.

Объединение модулей объектного кода в единую исполняемую программу является результатом работы компоновщика.

Процесс трансляции состоит из этапов:

1. Производится лексический анализ текста с разбиением его на элементы.
2. Производится синтаксический разбор.
3. Исходный код преобразуется в исполняемый машинный код.

Синтаксис языка программирования – это описание языка программирования, определяющее вид, форму и структуру конструкций языка.

Семантика языка программирования – это интерпретация абстрактного синтаксиса в терминах модели.

Классификация языков программирования

В настоящее время в мире существует несколько сотен реально используемых языков программирования. Для каждого есть своя область применения. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования – чем язык менее детален, тем выше его уровень. По этому критерию можно выделить следующие уровни языков программирования (рис. 8):

- **машинно-ориентированные языки** – машинные языки и языки ассемблера;
- **машинно-независимые языки** – языки высокого уровня.

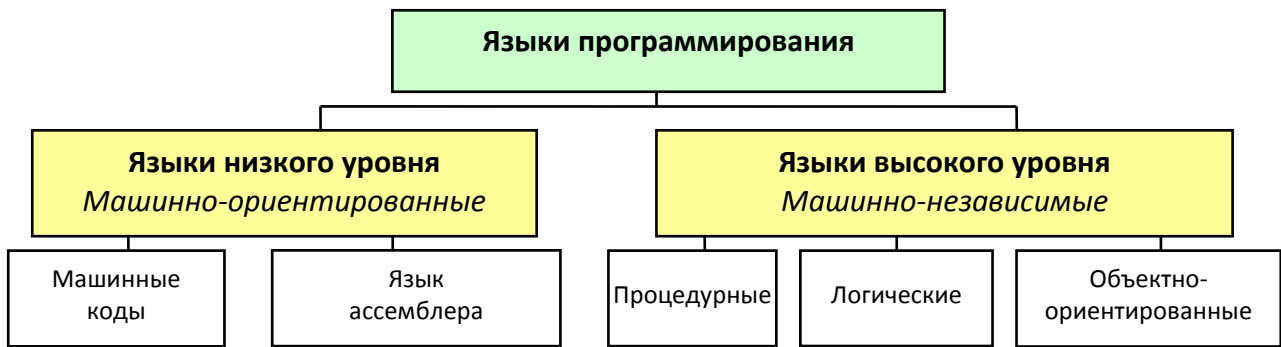


Рис. 8. Классификация языков программирования

Машинно-ориентированные языки – это языки низкого уровня, требующие указания мелких деталей процесса обработки данных. Языки же высокого уровня имитируют естественные языки, используя некоторые слова естественного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делятся на процедурные, логические и объектно-ориентированные.

- **Процедурные** или **алгоритмические** языки (Basic, Pascal, C и др.) предназначены для однозначного описания алгоритмов в виде некоторой последовательности операторов языка.
- **Логические** языки (Prolog, Lisp и др.) ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи, из которого должно следовать решение.
- **Объектно-ориентированные** языки (Pascal, C++, Java, C# и др.), в основе которых лежит понятие объекта, сочетают в себе данные и действия над ними. Программа на объектно-ориентированном языке, решая некоторую задачу, по сути, описывает часть мира, относящуюся к этой задаче. Описание действительности в форме системы взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

3. Основные объекты языка программирования QBASIC.

Простейшие конструкции языка

Основные конструкции записываются с помощью алфавита.

Алфавит языка включает в себя:

- прописные и строчные буквы латинского алфавита A – Z (a – z)
- цифры от 0 до 9
- знаки арифметических операций +, –, *, /, ^
- знаки операций отношений >, <, >=, <=, =, <>
- специальные знаки ! ? # (диез) % & (амперсанд) \$ “ . , ; : \ (слэш) ()
- буквы русского алфавита А – я – только для комментария

Константы.

Используют три типа констант: *целые*, *вещественные* и *строковые*.

Целая константа представляет собой запись числа, перед которой может стоять знак + или – без десятичной точки.

Вещественная константа. Это набор из одной или нескольких десятичных цифр со знаком и десятичной точкой. Вещественные константы могут быть представлены с различной точностью: *обычной* (6 знаков) и *двойной* (16 знаков).

Допускаются следующие формы записи вещественных констант с *обычной точностью*:

- *основная (естественная) форма записи числа с фиксированной точкой*.

Например, 73.249; –0.57; 13.2498.

- *экспоненциальная (показательная) форма записи числа с плавающей точкой*.

При записи константы в экспоненциальной форме число состоит из мантиссы, символа E (основания десятичной системы счисления) и порядка. Мантисса записывается по правилам записи констант, представленных или в основной форме, или целыми; порядок – по правилам записи целых констант с количеством цифр не более двух.

Например, константу $23,4 \cdot 10^{-2}$ можно записать в экспоненциальной форме в виде $23.4E - 2$ (где 23,4 – мантисса, –2 – порядок); $-18,94 \cdot 10^3 = -18,94E3$.

Эта форма записи констант удобна при записи очень больших либо очень малых чисел.

Числа, завершающиеся знаком ! , представляют собой вещественные константы.

Константы с удвоенной точностью имеют те же способы записи, что и константы с обычной точностью, только вместо буквы E для указания порядка употребляется D, а для завершающего знака вместо ! используется #.

Например, 0.54D0; 873,5#.

Строчковая константа – это последовательность любых алфавитно-цифровых и специальных символов длиной, не превышающей 255 символов; ограничиваются с двух сторон двойными кавычками.

Например, «ТАБЛИЦА F(X)», «ПРОГРАММА».

Переменные.

Переменная – эта величина, значение которой изменяется в процессе выполнения программы. Идентификатор (имя) переменной состоит из букв, цифр, точек и символа подчеркивания. Имя переменной – это произвольный набор символов, который может содержать от 1 до 40 символов. Первым символом имени должна быть буква.

Например: X, TABLE, A20RA, PERSON.FAM, SPISOK15

Язык Бейсик допускает три типа переменных: **вещественные, целые, строковые**. Для указания типа переменной к имени добавляется символ типа: % – целый, ! – с обычной точностью (6 знаков), # – с двойной точностью (16 знаков), \$ – строковый. По умолчанию переменная относится к вещественному типу с обычной точностью.

Например, B, X(I) – переменные вещественного типа;
 B%, X%(I) – переменные целого типа;
 B!, X!(I) – переменные вещественного типа с обычной точностью;
 B#, X#(I) – переменные вещественного типа с удвоенной точностью;
 B\$, X\$(I) – переменные строкового типа.

Различают переменные *простые* и переменные *с индексами*.

Простая переменная определяется только идентификатором.

Переменная с индексом является элементом массива и определяется идентификатором и индексом, значения которого обозначает номер элемента в

массиве. Индекс записывается в круглых скобках и может представляться числом, переменной или арифметическим выражением и принимать только целые положительные значения или нуль.

Например, $X(5)$ – пятый элемент числового массива X ;

$Z(I, J)$ – элемент i -ой строки и j -го столбца матрицы Z ;

$X(I+2)$ – $(I+2)$ -й элемент массива X .

Допускается использование одинаковых имен для простой переменной и переменной с индексом.

Например, Z и $Z(J)$

Функции.

Функция – вид подпрограммы, вычисляющий единственное значение по исходным данным, причем при одних и тех же входных данных, всегда получается один результат. Стандартные функции используются для вычисления часто употребляемых функций. Аргумент стандартной функции заключается в круглые скобки. В качестве аргумента можно употреблять любое арифметическое выражение, число, переменную. В таблице 2 представлены примеры списка функций для обработки числовых данных, включенных в интерпретатор языка QBasic.

Таблица 2. **Функции для обработки числовых данных**

Математическое обозначение функции	QBASIC
$\sin x$	SIN(X)
$\cos x$	COS(X)
$\operatorname{tg} x$	TAN(X)
$\operatorname{arctg} x$	ATN(X)
$\sqrt{\delta}$	SQR(X)
e^x	EXP(X)
$ x $	ABS(X)
$\ln x$	LOG(X)
$\lg x$	LOG10(X)

$\log_a x = \frac{\ln x}{\ln a}$	LOG(X)/LOG(A)
$\lg x = \frac{\ln x}{\ln 10}$	LOG(X)/LOG(10)
$\text{ctgx} = \frac{1}{\text{tgx}}$	1/TAN(X)

Выражения.

Выражение – это последовательная запись констант, переменных, функций или любых комбинаций, образованная с помощью знаков арифметических операций, операций отношения, операций над строками символов. Различают выражения *арифметические*, *строковые* и *логические*.

Арифметические выражения. Результатом вычисления арифметического выражения является число.

Арифметические операции:

- | | | | |
|---|-------------|-----|------------------------|
| + | (сложение) | ^ | (возведение в степень) |
| - | (вычитание) | \ | (деление нацело) |
| * | (умножение) | MOD | (определение остатка) |
| / | (деление) | DIV | (выделение целого) |

При делении нацело дробная часть результата отбрасывается, так $5 \setminus 2 = 2$

Операция вычисления остатка (деления по модулю) определяет остаток от деления одного целого числа на другое, так $10 \text{ MOD } 3$ есть 1.

Операция выделение целого определяет целую часть от деления одного целого числа на другое, так $10 \text{ DIV } 3$ есть 3.

При записи выражения нельзя опускать знак операции умножения, ставить рядом два знака операций.

Таблица 3. *Примеры записи выражений*

Математическая запись	QBASIC
-----------------------	--------

$3a^2 + \frac{b}{4} + \frac{7}{1-a}$	$3*a^2+b/4+7/(1-a)$
$\frac{a - \sqrt{d}}{2,5x}$	$(a-SQR(d))/(2.5*x)$
$\sin^2 x - \cos x^2$	$SIN(x)^2-COS(x)^2$

Порядок выполнения операций в выражении задается установленным приоритетом операций: *вычисление функции, возведение в степень, умножение или деление, деление нацело, определение остатка, сложение или вычитание.*

Для задания нужной последовательности действий используются круглые скобки.

Строковые выражения. Строковое выражение состоит из строковых констант, строковых переменных и строковых функций. Результатом вычисления строкового выражения является строка символов. В строковых выражениях используется операция-сцепление (*конкатенация*) строковых значений, обозначаемая знаком +.

Например: A\$+B\$ означает присоединение строки B\$ к концу строки A\$. Если A\$=«Иванов» B\$=«группа Э-11», то результат операции A\$+B\$ есть строковое значение «Иванов группа Э-11». Результирующая строка не должна превышать 255 символов.

Логические выражения. Простейшим логическим выражением является отношение, которое применяется для сравнения значений арифметических или строковых выражений. Допустимы шесть типов отношений: = (равно), < > (не равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно).

Результатом отношения является логическое значение «истина» – если условие отношения удовлетворяется, или «ложь» – в противном случае. Отношения используются в программах для организации разветвлений вычислительного процесса.

Таблица 4. **Примеры записи отношений**

Математическая запись	QBASIC
$b^2 - 4ac > 0$	$b^2 - 4 * a * c > 0$
$i \neq j$	$i < > j$
$x_{ij} \leq y$	$x(i, j) \leq y$
строка a = строке b	$a\$ = b\$$

Если сравнения выполняются над числовыми значениями, то допустимы сравнения значений целого и вещественного типов. Строковые значения можно сравнить только с другими строковыми значениями. Сравнение строк производится посимвольно слева направо. При сравнении символьных строк разной длины короткая строка считается меньшей. При несовпадении символов большей считается строка, содержащая символ с большим значением кода символа в таблице кодов.

Возможно использование составных отношений, образованных с помощью символов логических операций: **NOT** – отрицание, **AND** – умножение (логическое И), **OR** – сложение (логическое ИЛИ), **EQV** – эквивалентность, **IMP** – импликация, **XOR** – «исключающее или».

Таблица 5. *Примеры записи составных отношений*

Математическая запись	QBASIC
$-4 \leq x \leq 4$	$(-4 \leq x) \text{ AND } (x \leq 4)$
$a > 0$ и $b > 0$	$a > 0 \text{ AND } b > 0$
$x = 0$ или $x = 2$	$x = 0 \text{ OR } x = 2$

Основные операторы языка QBASIC

Минимальная конструкция алгоритмического языка, задающая полное описание некоторого действия, которое необходимо выполнить, называется *оператором*.

С помощью операторов описываются действия, которые необходимо выполнить для реализации алгоритма решения задачи.

Структура программы. Программа представляет собой последовательность операторных строк. Каждая строка может начинаться с номера строки,

который служит меткой оператора и используется для ссылки на него при передаче управления, а также определяет очередность выполнения операторов. Целесообразно использовать номера, кратные 10, чтобы иметь возможность вставлять при отладке дополнительные строки.

Метками могут быть не только номера, но и набор алфавитно-цифровых символов и символов подчеркивания.

В строке помещают один или несколько операторов (многооператорная строка). Между операторами одной строки ставится разделительный знак « : » (двоеточие).

При указании на конец вычислений последним оператором программы должен быть оператор **END** – оператор конца программы.

Программы должны быть понятны не только компьютеру, но и человеку, не знакомому с данной программой. Поэтому желательно в самой программе коротко комментировать ее назначение, алгоритм решения. Для этого используется оператор-комментарий **REM**.

Формат оператора REM:

nc REM текст

nc – номер строки (можно не указывать);

текст – текст комментария, содержащий любые символы из алфавита языка и буквы русского алфавита.

Ключевое слово REM может быть заменено апострофом.

Каждую программу желательно начинать комментарием с указания имени программы, ее автора, даты разработки, назначения.

Например,

10 REM Программу составил студент Иванов

10 ' Программу составил студент Иванов

Оператор присваивания LET служит для вычисления значения арифметического или строкового выражения и присваивания этого значения переменной.

Формат оператора LET:

nc LET x = a

LET – необязательное имя оператора присваивания;

x – имя переменной, принимающей новое значение;

a – выражение.

Переменная и выражение должны быть или оба числовыми, или оба строковыми.

Например,

```
10 LET A=4,5
```

Оператор предписывает присвоить значение 4,5 числовой переменной А.

Оператор CLS – предназначен для очистки экрана.

QBASIC не очищает автоматически экран при запуске новой программы, поэтому рекомендуется начинать программу с оператора CLS.

Операторы ввода данных.

Оператор ввода состоит из служебного слова, определяющего характер действия ЭВМ над числами, переменными, выражениями. Следом за служебным словом располагается так называемое тело оператора. Операторы ввода служат для задания исходных данных при выполнении программ.

Для ввода данных используется оператор **INPUT**.

Формат оператора INPUT:

```
nc INPUT список переменных
```

При выполнении оператора INPUT на экран монитора выводится знак вопроса и выполнение программы приостанавливается. После этого пользователь вводит с клавиатуры значение каждой переменной, указанной в списке, по порядку, разделяя их запятыми (целая часть от дробной отделяется точкой). После ввода последнего значения запятая не ставится, а нажимается клавиша Enter и продолжается выполнение программы.

Например,

```
20 INPUT a, b, c
```

```
? 1, 2.5, -3
```

Число вводимых данных должно соответствовать числу переменных в операторе INPUT.

При работе в режиме диалога с компьютером не всегда удобно, чтобы запрос на ввод с клавиатуры тех или иных значений вводился лишь в виде одного знака вопроса. Необходимо предусмотреть в программе вывод на экран наводящих сообщений («подсказка») о том, какую информацию следует ввести пользователю.

Оператор позволяет кроме знака вопроса (или вместо него) выводить на экран текстовые наводящие сообщения. Текст такого сообщения должен указы-

ваться в операторе INPUT в виде строковой константы, стоящей непосредственно за ключевым словом INPUT, при этом строковая константа заключается в кавычки и отделяется от имени переменной точкой с запятой.

Например,

20 INPUT "Введите 3 числа a, b, c"; a, b, c

на экране появится сообщение

Введите 3 числа a, b, c?

Пользователь дополняет строку необходимой информацией.

Введите 3 числа a, b, c? 1, 2.5, -3

Для ввода данных могут также использоваться операторы **READ** и **DATA**, которые выполняются совместно: **DATA** хранит данные, а **READ** к ним обращается.

Формат операторов READ и DATA:

nc READ список переменных

nc DATA список констант

В списке переменных перечисляются через запятые переменные, а *в списке констант* – любые допустимые числовые или строковые значения этих переменных. Строковая константа заключается в кавычки.

Количество переменных в операторах READ и констант в операторах DATA должно совпадать. Однако операторы READ и DATA не обязательно должны быть парными. *Например*, двум операторам READ может соответствовать один оператор DATA и наоборот.

Пример. Необходимо присвоить значения переменным: a=1, b=2.5, c=-3, d="Иванов".

Это можно выполнить разными способами:

1. Один оператор READ и один оператор DATA

10 DATA 1, 2.5, -3, Иванов

20 READ a, b, c, d\$

2. Два оператора READ и один оператор DATA

10 DATA 1, 2.5, -3, Иванов

20 READ a, b

30

80 READ c, d\$

3. Два оператора READ и два оператора DATA

10 DATA 1, 2.5

20 READ a, b

30

80 DATA -3, Иванов

90 READ c, d\$

Если в программе используется несколько операторов DATA, то все они участвуют в создании одного и того же блока данных.

Перед выполнением программы QBasic просматривает все операторы DATA в порядке возрастания номеров строк и создает единый блок данных. Если в программе встречается оператор READ, то считывается текущее значение констант из блока данных и запоминается положение последнего считанного данного (с помощью указателя позиции). Следующий оператор READ начинает выбирать данные с той позиции, которая была установлена предыдущим оператором READ.

Неиспользованные значения из списка DATA игнорируются. Если блок данных исчерпан, то при попытке продолжить чтение данных выдается сообщение об ошибке.

Операторы DATA относятся к числу неисполняемых операторов. Их можно размещать в любой части программы до оператора END. Однако рекомендуется избегать разбрасывания операторов DATA по программе. Лучше всего их помещать в одном месте программы – в начале или в конце.

Оператор восстановления данных **RESTORE**. Оператор позволяет повторно вновь выбрать из оператора DATA уже перед этим выбранные данные. После его выполнения выборка из оператора DATA начинается с самого первого числа.

Формат оператора:

nc RESTORE

Например,

10 DATA 1, 2, 3, 4

20 READ a, b

30 RESTORE

40 READ c, d, e, f

Первый оператор READ (строка 20) читает первые два значения из оператора DATA (строка 10):

a=1 b=2

Затем оператор RESTORE (строка 30) перемещает указатель в начало блока данных (строка 10). Второй оператор READ (строка 40) снова читает теперь уже четыре числа:

c=1 d=2 e=3 f=4

Оператор вывода данных PRINT используется для вывода данных на экран монитора в процессе выполнения программы.

Формат оператора:

nc PRINT список вывода

Список вывода может содержать константы, переменные, строковые или арифметические выражения, функции. Элементы списка разделяются запятыми или точкой с запятой.

Например,

30 PRINT X, Y – вывод значений числовых переменных X и Y

40 PRINT A\$ – вывод значения строковой переменной A

50 PRINT "результаты вычислений" – вывод указанного в кавычках текста

60 PRINT 3+4 – вывод результата сложения двух чисел, равный 7

Оператор PRINT без списка аргументов выводит строку пробелов.

При использовании оператора PRINT можно организовать вывод данных в двух форматах – *ЗОННОМ* и *КОМПАКТНОМ*.

При *зонном формате* строка разделяется на 5 зон по 14 позиций в каждой. Каждое выводимое значение размещается в своей зоне. Если последняя зона в строке заполнена, то вывод данных продолжается с первой зоны следующей строки. При выводе числовых констант первая позиция каждой зоны отводится под знак числа. Для вывода данных в этом формате необходимо элементы списка вывода отделять друг от друга *запятыми*. Наличие в списке оператора PRINT двух рядом стоящих запятых вызывает пропуск одной зоны.

Например,

10 a = 1 : b = 2 : c = -3 : d = 146 : e = -55

```

20 PRINT a, b, c, d, e
30 a = -1 : c = 5 : d$ = "ПРОГРАММА"
40 PRINT a,, b,, c, d$

```

на экране будет выведено

```

1      2      -3      146      -55
-1          2          5
ПРОГРАММА

```

При *компактном формате* числовые значения выводятся через две или одну позицию (если перед числом стоит знак минус), а строковые значения – подряд без пробелов. Для организации вывода в таком формате необходимо отделить элементы списка вывода друг от друга *точкой с запятой*.

Например,

```

10 a = 1 : b = 2 : c = -3 : d = 146 : e = -55
20 PRINT a; b; c; d; e
30 PRINT "СУММА=", a + d
40 PRINT "СУММА="; a + d

```

на экране будет выведено

```

1  2  -3  146  -55
СУММА=          147
СУММА=147

```

Расположение операторов при выводе, кроме разделителей, может определяться специальной функцией `TAB(n)`.

Где n – номер позиции, с которой должен выводиться элемент, стоящий за данной функцией.

Например, оператор

```
70 PRINT TAB(30); "ГРАФИК"; TAB(50); "Y=F(X)"
```

напечатает слово `ГРАФИК`, начиная с 30-й позиции текущей строки, а `Y=F(X)` – с 50-й позиции.

Операторы безусловного и условного переходов – данные операторы используются для изменения последовательности выполнения операторов.

Оператор безусловного перехода `ГOTO n` употребляется для перехода к выполнению строки с номером n .

Например, при выполнении оператора

40 GOTO 10 – будет осуществлен переход к оператору
в строке с меткой 10.

Оператор GOTO должен быть либо единственным оператором строки, либо последним оператором многооператорной строки.

Оператор условного перехода **IF THEN** и **IF GOTO** изменяет последовательность выполнения операторов программы в зависимости от значения проверяемого условия.

Формат оператора:

nc **IF** *условие* **GOTO** *номер строки n* [**ELSE** *оператор*] (1)

nc **IF** *условие* **THEN** *оператор 1* [**ELSE** *оператор 2*] (2)

где *условие* – выражение типа отношения (=, <, >, <=, >=, <>); *оператор* – любой выполняемый оператор или группа операторов, разделенных двоеточием.

Часть оператора, начиная с ключевого слова ELSE, может быть опущена.

Если результат отношения «истина», то управление передается на строку с номером *n* (формат записи 1) либо выполняется оператор 1 или группа операторов, записанных после ключевого слова THEN (формат записи 2). Если результат отношения «ложь», то выполняется оператор 2 или группа операторов из ветви ELSE. Если ветвь ELSE отсутствует, то управление передается строке, следующей за IF.

Например,

```
20 IF x > 0 GOTO 50
```

Если условие $x < y$ выполняется, то осуществляется переход на строку 50.

Например,

```
20 IF x < y THEN x = y
```

При выполнении условия переменной *x* будет присвоено значение переменной *y*, если условие не будет выполнено, то произойдет переход к следующей строке.

В приведенном ниже фрагменте программы

```
10 INPUT "Укажите свой возраст";K
```

```
20 IF K>=18 THEN PRINT "Вы можете голосовать"
```

ELSE PRINT “Вы слишком молоды” : PRINT “ Вы можете
голосовать через ”18 – К;”лет”

30 END

оператор IF сравнивает введенное значение К с числом 18, которое означает возраст, допустимый для голосования. Если введенное значение $K \geq 18$, то выполняется оператор PRINT ветви THEN, а если $K < 18$, то выполняются операторы ветви ELSE.

Операторы цикла FOR NEXT позволяют выполнить тело цикла несколько раз с изменением параметра цикла от начального до конечного значения с постоянным шагом.

Формат операторов:

nc FOR $i = a_1$ TO a_2 STEP a_3

S_1
 S_2
...
 S_k

} Операторы тела цикла

nc NEXT i

где FOR – для; TO – до; STEP – шаг; NEXT – следующий; i – управляющая переменная, параметр цикла; a_1 и a_2 – начальное и конечное значения параметра цикла; a_3 – шаг изменения параметра цикла (a_1, a_2, a_3 – любые арифметические выражения); S_1, S_2, \dots, S_k – любые выполняемые операторы.

Оператор FOR NEXT используется только в паре. Оператор FOR определяет начало цикла, NEXT – конец цикла. Управляющая переменная i , указанная в операторе FOR, обязательно должна совпадать с переменной i , указанной в операторе NEXT.

По умолчанию шаг изменения параметра цикла a_3 принимается равным 1.

Порядок выполнения операторов цикла следующий:

- 1) вычисляются значения арифметических выражений a_1, a_2, a_3 ;
- 2) параметру цикла i присваивается начальное значение a_1 ;

3) если шаг $a_3 > 0$ и $i \leq a_2$ или $a_3 < 0$ и $i \geq a_2$, то выполняются операторы, образующие тело цикла. Значение параметра цикла увеличивается на величину шага $i = i + a_3$ и повторяется п. 3.

В противном случае осуществляется выход из цикла и управление передается оператору, следующему за NEXT.

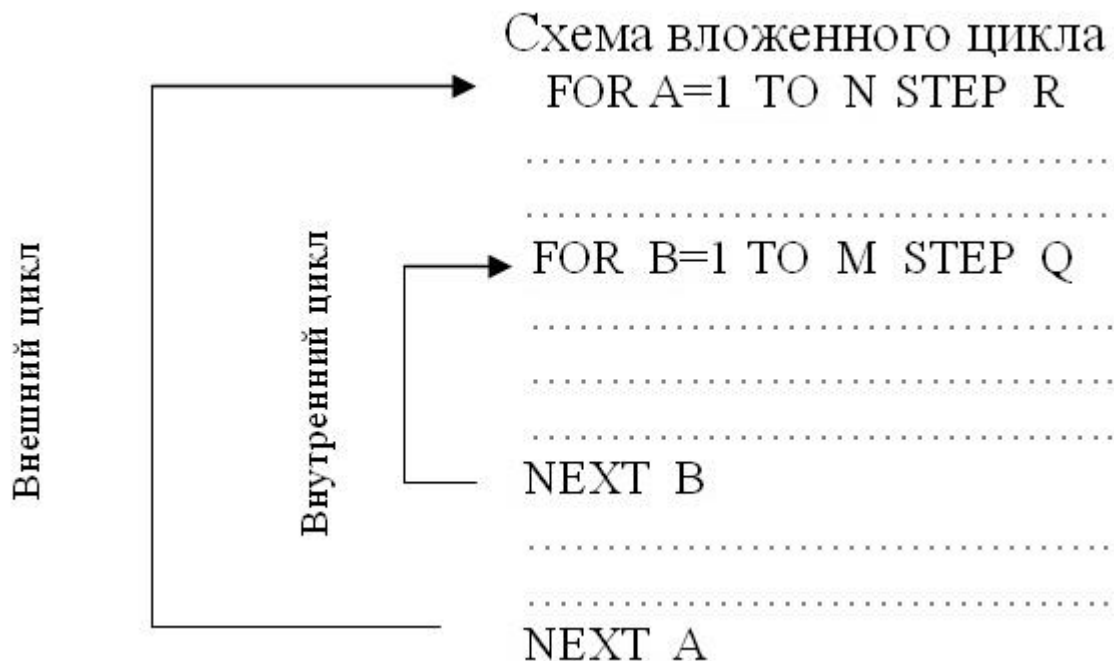
Количество повторений цикла определяется по формуле:

$$n = \frac{a_2 - a_1}{a_3} + 1$$

Например, необходимо ввести 10 значений величины X и вычислить корень квадратный.

```
10 FOR I=1 TO 10
20 INPUT X
30 PRINT SQR (X)
40 NEXT I
```

С помощью операторов FOR и NEXT в QBASIC возможно осуществить организации циклов внутри цикла. При этом внутренний цикл называется вложенным. И он полностью должен находиться внутри внешнего цикла.



Как правило, внутренних (вложенных) циклов может быть столько, сколько позволяет ЭВМ. Главное, чтобы они строго находились один внутри другого.

Программа с вложенным циклом имеет вид:

```
10  FOR A=1 TO 10
20  FOR B=1 TO 5
30  INPUT C(a, b)
40  NEXT B
50  NEXT A
```

Циклы с предусловием лучше организовывать, используя операторы **WHILE WEND** – первым из них начиная цикл, а вторым завершая его.

Общий вид оператора **WHILE**:

```
WHILE условие
```

Оператор **WEND** задается без операндов. Порядок выполнения такого цикла: проверяется выполнение условия в операторе **WHILE**. Если его значение «истина», то выполняются операторы тела цикла, расположенные до оператора **WEND**.

Работа цикла завершается, когда значение проверяемого условия будет «ложь». В этом случае управление передается оператору, следующему за **WEND**.

С использованием операторов **WHILE** и **WEND** программа будет иметь вид:

```
20 INPUT a, x
30 k = 1 REM начальное значение параметра
40 WHILE x ^ k / k > a
50 k = k + 1
60 WEND
70 PRINT k : END
```

Обработка массивов.

Массив – упорядоченный набор переменных одного типа, для хранения которых отводятся последовательно расположенные поля памяти. Массив характеризуется идентификатором, числом измерений (индексов) и числом переменных в каждом измерении (верхней границей каждого индекса).

Для образования имен массивов используются те же правила, что и для образования имен простых переменных. В одной и той же программе допустимы одинаковые имена массивов и простых переменных. В языке Бейсик воз-

можно работа с одномерными массивами (векторы, линейные таблицы), двумерными массивами (матрицы, прямоугольные таблицы).

При резервировании места в памяти в программе необходимо указать для каждого массива максимальный размер, т.е. максимальные значения каждого из индексов. Информация о размерности массива содержится в операторе **DIM**, имеющий формат

пс DIM список массивов

Список массивов содержит имена массивов и в круглых скобках разделенные запятыми верхние границы каждого из индексов.

Так, оператор DIM A (10) резервирует место в памяти для одномерного массива A, состоящего из 11 элементов – от A (0) до A (10). Оператор DIM B(6, 5) резервирует место под двумерный массив B, содержащий семь строк и шесть столбцов (включая нулевые).

В операторе DIM для определения размера массива могут использоваться только целые константы. Минимальным значением индекса может быть 0 либо 1.

Обработка массивов ведется поэлементно. Все операции над массивом осуществляются в цикле.

Например, ввод одномерного массива A содержащего 20 элементов:

```
10 DIM A (20)
20 FOR I=1 TO 20
30 INPUT A (I)
40 NEXT I
```

Ввод двумерного массива E, состоящего из двух строк и трех столбцов (минимальное значение индекса равно 1).

```
10 DIM E (2, 3)
20 FOR I=1 TO 2
30 FOR J=1 TO 3
40 INPUT E (I, J)
50 NEXT J
60 NEXT I
```

Различают выполняемые и невыполняемые операторы.

Невыполняемые операторы описывают свойства данных, например, размерность массива (оператор DIM), комментируют текст программы (опера-

тор REM). Невыполняемые операторы могут располагаться в любом месте программы до оператора END.

Выполняемые операторы реализуют действия, предусмотренные алгоритмом решения задачи, например, присвоить значение, выдать на печать, передать управление, остановить процесс вычислений и т.д. Среди выполняемых операторов можно выделить операторы: присваивания, ввода-вывода, операторы перехода.

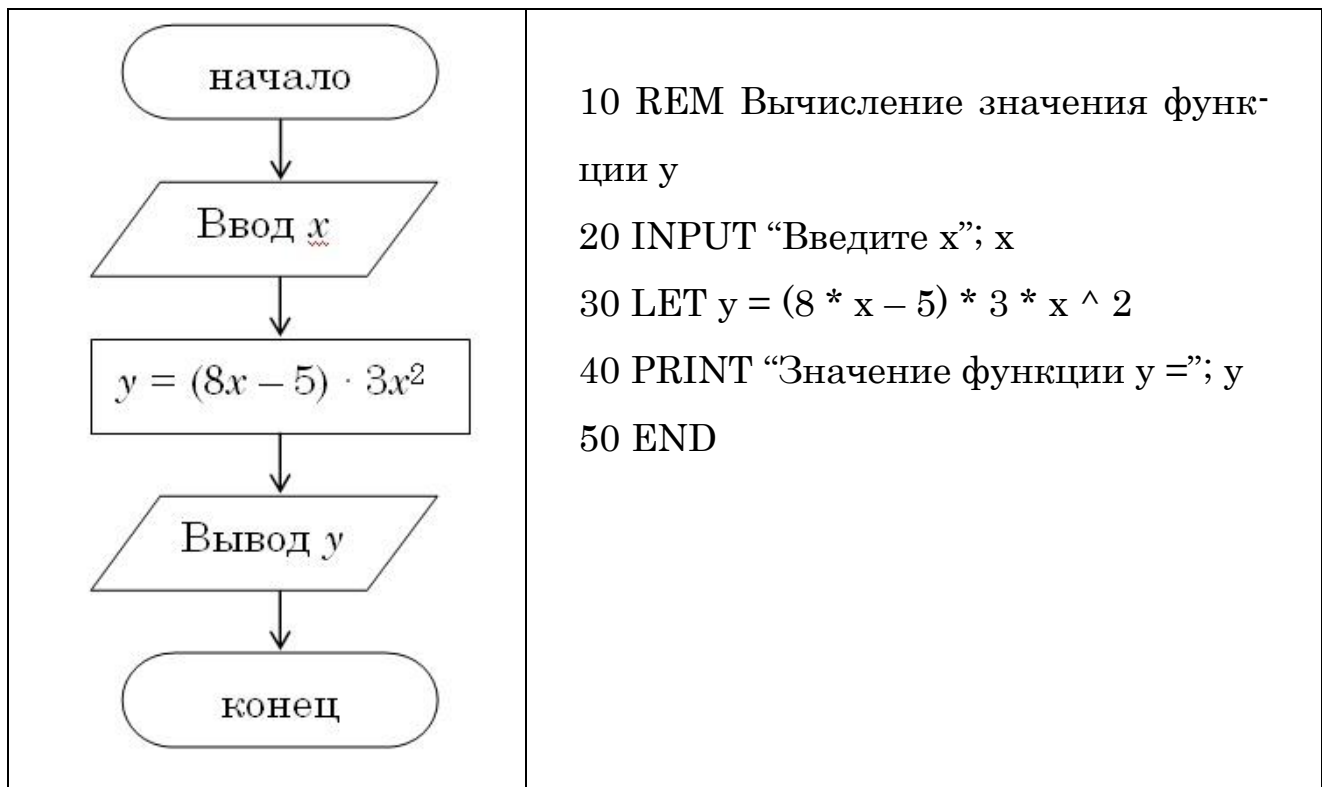
Более подробную информацию о функциях и операторах QBasic можно получить в программе с помощью Справки, используя меню или клавишу F1.

4. Примеры программирования на языке QBASIC.

Программирование линейного вычислительного процесса

Пример. Вычислить значение функции $y = (8x - 5) \cdot 3x^2$.

Блок-схема	QBasic
------------	--------



Программирование разветвляющегося вычислительного процесса

Пример. Вычислить значения функции z в зависимости от условий:

$$z = \begin{cases} x - y, & \text{если } x > y \\ y - x, & \text{если } x \leq y \end{cases}$$

$$z = \begin{cases} x - y, & \text{если } x > y \\ y - x, & \text{если } x < y \\ x + y, & \text{если } x = y \end{cases}$$

Блок-схема	QBasic
------------	--------

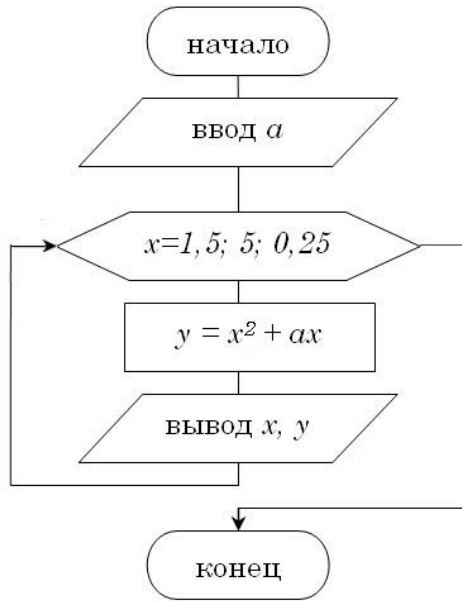
<pre> graph TD Start([начало]) --> Input[/Ввод x, y/] Input --> Decision{x > y} Decision -- да --> Process1[z = x - y] Decision -- нет --> Process2[z = y - x] Process1 --> Output[/Вывод z/] Process2 --> Output Output --> End([конец]) </pre>	<pre> 10 REM Вычисление функции z 20 PRINT "Ввести x, y" 30 INPUT x, y 40 IF x > y THEN z = x - y 50 z = y - x 60 PRINT "Значение z ="; z 70 END </pre>
<pre> graph TD Start([начало]) --> Input[/Ввод x, y/] Input --> Decision1{x > y} Decision1 -- да --> Process1[z = x - y] Decision1 -- нет --> Decision2{x < y} Decision2 -- да --> Process2[z = y - x] Decision2 -- нет --> Process3[z = x + y] Process1 --> Output[/Вывод z/] Process2 --> Output Process3 --> Output Output --> End([конец]) </pre>	<p><i>Программу для данной блок-схемы напишите самостоятельно</i></p>

Программирование циклического вычислительного процесса

Пример. Вычислить таблицу значений функции $y = x^2 + ax$ при изменении x в пределах от 1,5 до 5 с шагом 0,25.

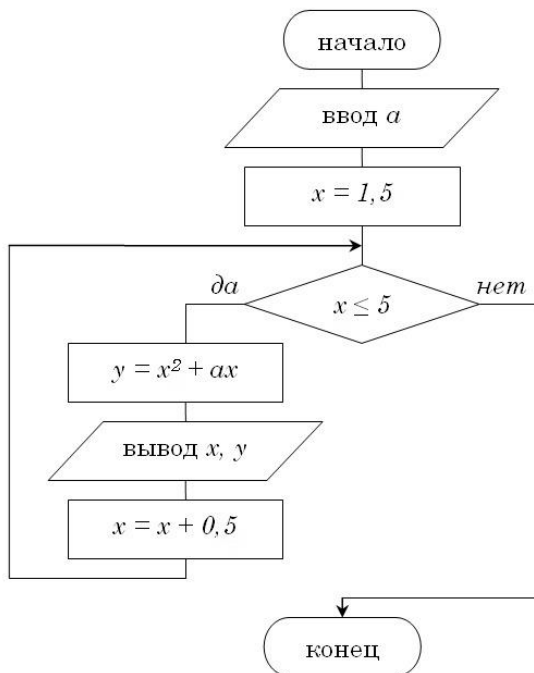
Блок-схема	QBasic
------------	--------

Цикл с параметром



```
10 REM Табулирование функции y
20 PRINT "Ввести значение a"
30 INPUT a
40 FOR x = 1.5 TO 5 STEP 0.25
50 y = x ^ 2 + a * x
60 PRINT "x = "; x, "y = "; y
70 NEXT x
80 END
```

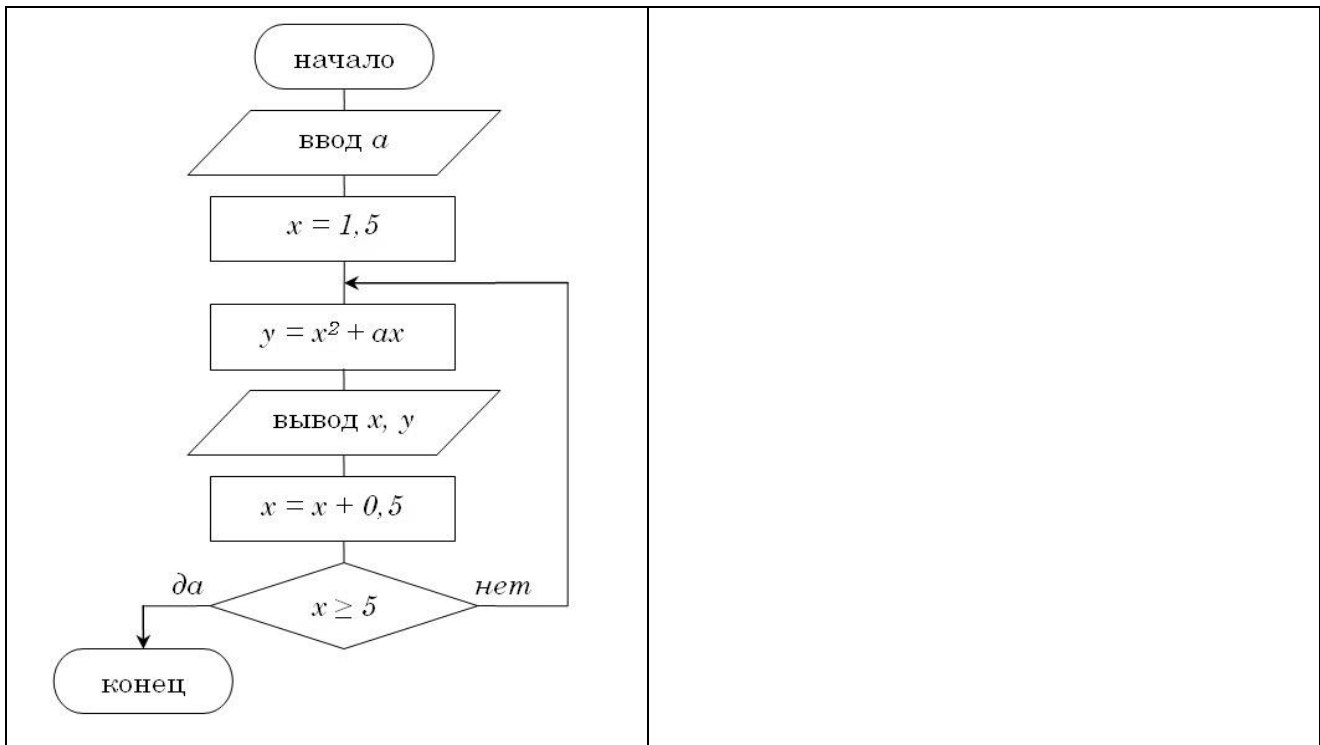
Цикл с предусловием



Программу для данной блок-схемы напишите самостоятельно

Цикл с постусловием

Программу для данной блок-схемы напишите самостоятельно



Контрольные вопросы

1. Кто и когда впервые ввел понятие алгоритма?
2. Какие из нижеперечисленных правил являются алгоритмами?
 Ответ обоснуйте:
 - орфографические правила;
 - правила выполнения арифметических операций;
 - правила техники безопасности;
 - правила перевода чисел из одной системы счисления в другую.
3. Что представляет собой графическая форма записи алгоритма?
4. Что такое итерация? Какие циклы называют итерационными?
5. В чем состоит различие между естественными языками и языками программирования?
6. С какими видами данных можно работать на языке QBASIC?
7. Из каких символов можно составить имя переменной?
8. Чем компиляция отличается от интерпретации?
9. Какие вы знаете классификации языков программирования?
10. Какие языки и системы программирования вы знаете и в чем их особенности?

Лекция 8.

ОСНОВЫ МОДЕЛИРОВАНИЯ

1. Моделирование как метод познания.
2. Формы представления моделей. Формализация.
3. Типы информационных моделей.

1. Моделирование как метод познания.

Моделирование. Человечество в своей деятельности (*научной, образовательной, технологической, художественной*) постоянно создает и использует модели окружающего мира. Строгие правила построения моделей сформулировать невозможно, однако накоплен богатый опыт моделирования различных объектов и процессов.

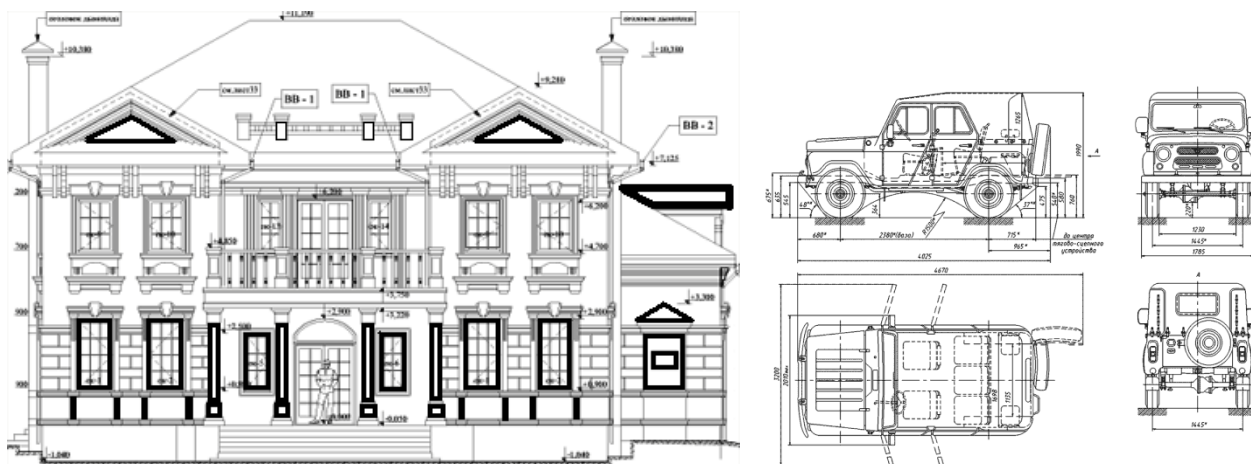
Оригиналы и модели



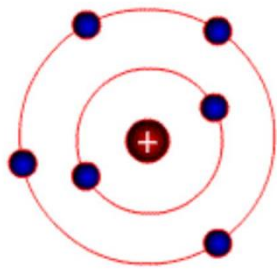
Модели позволяют представить **в наглядной форме** объекты и процессы, недоступные для непосредственного восприятия (*очень большие или очень маленькие объекты, очень быстрые или очень медленные процессы и пр.*). Наглядные модели часто используются в процессе обучения. В курсе геогра-

фии первые представления о нашей планете Земля получаем, изучая ее модель – глобус, в курсе физики изучаем работу двигателя внутреннего сгорания по его модели, в химии при изучении строения вещества используем модели молекул и кристаллических решеток, в биологии строение человека по анатомическим муляжам и др.

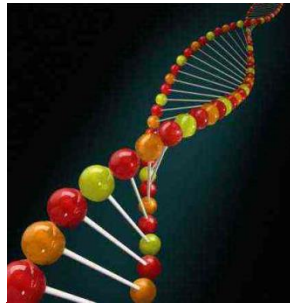
Модели играют важную роль **в проектировании** и создании различных технических устройств, машин и механизмов, зданий, электрических цепей и т.д. Без предварительного создания чертежа невозможно изготовить даже простую деталь, не говоря уже о сложном механизме. В процессе проектирования зданий и сооружений кроме чертежей изготавливают макеты. Разработка электрической схемы обязательно предшествует созданию электрических цепей и т.д.



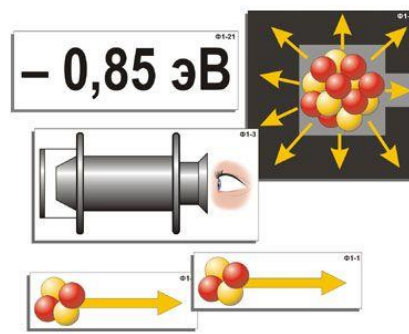
Развитие науки невозможно без создания **теоретических моделей** (теорий, законов, гипотез и пр.) отражающих строение, свойства и поведение реальных объектов. Создание новых теоретических моделей иногда коренным образом меняет представление человечества об окружающем мире (*гелиоцентрическая система мира Коперника, модель атома Резерфорда-Бора, модель расширяющейся Вселенной, модель генома человека* и пр.). Адекватность теоретических моделей законам реального мира проверяется с помощью опытов и экспериментов.



МОДЕЛЬ АТОМА УГЛЕРОДА



МОДЕЛЬ ГЕНОМА ЧЕЛОВЕКА



МОДЕЛЬ АТОМА
РЕЗЕРФОРДА-БОРА

Все *художественное творчество* фактически является процессом создания моделей. Например, такой литературный жанр, как басня, переносит реальные отношения между людьми на отношения между животными и фактически создает модели человеческих отношений. Более того, практически любое литературное произведение может рассматриваться как модель реальной человеческой жизни. Моделями, в художественной форме отражающими реальную действительность, являются также живописные полотна, скульптуры, театральные постановки и пр.



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

Модель. Каждый объект имеет большое количество различных свойств. В процессе построения модели выделяются главные, наиболее существенные для проводимого исследования свойства.

Разные науки исследуют объекты и процессы под разными углами зрения и строят различные типы моделей. В физике изучаются процессы взаимодействия и изменения объектов, в химии – их химический состав, в биологии – строение и поведение живых организмов и так далее.

Возьмем в качестве примера человека: в разных науках он исследуется в рамках различных моделей. В рамках механики его можно рассматривать как материальную точку, в химии – как объект, состоящий из различных химических веществ, в биологии – как систему, стремящуюся к самосохранению, и так далее.

Модель – это некий новый объект, который отражает существенные особенности изучаемого объекта, явления или процесса.

География, военное дело, судоходство невозможны без информационных моделей поверхности Земли в виде карт. Различные типы географических карт (политические, физические и прочие) представляют информационные модели, отражающие различные особенности земной поверхности, то есть *один и тот же объект отражают несколько моделей*.

С другой стороны, *разные объекты могут описываться одной моделью*. Так в механике различные материальные тела (от планеты до песчинки) могут рассматриваться как материальные точки.

Один и тот же объект может иметь **множество моделей**, а разные объекты могут описываться **одной моделью**.

Никакая *модель не может заменить сам объект*. Но при решении конкретной задачи, когда интересуют определенные свойства изучаемого объекта, модель оказывается полезным, а подчас и единственным инструментом исследования.

2. Формы представления моделей. Формализация.

Предметные и информационные модели. Все модели можно разбить на два больших класса: *модели предметные (материальные)* и *модели информационные*.

Предметные модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме (*глобус, анатомические муляжи, модели кристаллических решеток, макеты зданий и сооружений и др.*).

Информационные модели представляют объекты и процессы в *образной* или *знаковой* форме.

Образные модели (*рисунки, фотографии*) представляют собой зрительные образы объектов, зафиксированные на каком-либо носителе информации (бумаге, фото- и киноплёнке и др.). Широко используются образные модели в образовании (*учебные плакаты по различным предметам*) и науках, где требуется классификация объектов по их внешним признакам (*в ботанике, биологии, палеонтологии и др.*).

Знаковые модели строятся с использованием различных языков (знаковых систем). Знаковая информационная модель может быть представлена в форме текста (*например, программа на языке программирования*), формулы (*например, второго закона Ньютона $\bar{F} = m \cdot \bar{a}$*), таблицы (*например, периодической таблицы элементов Д.И. Менделеева*) и т.д. Иногда при построении знаковых информационных моделей используются одновременно несколько различных языков. Примерами таких моделей могут служить географические карты, графики, диаграммы и пр. Во всех этих моделях используются одновременно как язык графических элементов, так и символьный язык.

На протяжении своей истории человечество использовало различные способы и инструменты для создания информационных моделей. Эти способы постоянно совершенствовались. Так, первые информационные модели создавались в форме наскальных рисунков, в настоящее же время информационные модели обычно строятся и исследуются с использованием современных компьютерных технологий.

Формализация. Естественные языки используются для создания *описательных информационных моделей*. В истории науки известны многочисленные описательные информационные модели, *например, гелиоцентрическая модель мира*, которую предложил Коперник, формулировалась следующим образом:

- Земля вращается вокруг своей оси и вокруг Солнца;
- орбиты всех планет проходят вокруг Солнца.

С помощью формальных языков строятся *формальные информационные модели* (математические, логические и др.). Одним из наиболее широко используемых формальных языков является математика. Модели, построенные с использованием математических понятий и формул, называются *математическими моделями*. Язык математики является совокупностью формальных языков.

Язык алгебры логики позволяет строить *формальные логические модели*. С помощью алгебры логики можно формализовать (записать в виде логических выражений) простые и сложные высказывания, выраженные на естественном языке. Построение логических моделей позволяет решать логические задачи, строить логические модели устройств компьютера (сумматора, триггера) и так далее.

Процесс построения информационных моделей с помощью формальных языков называется **формализацией**.

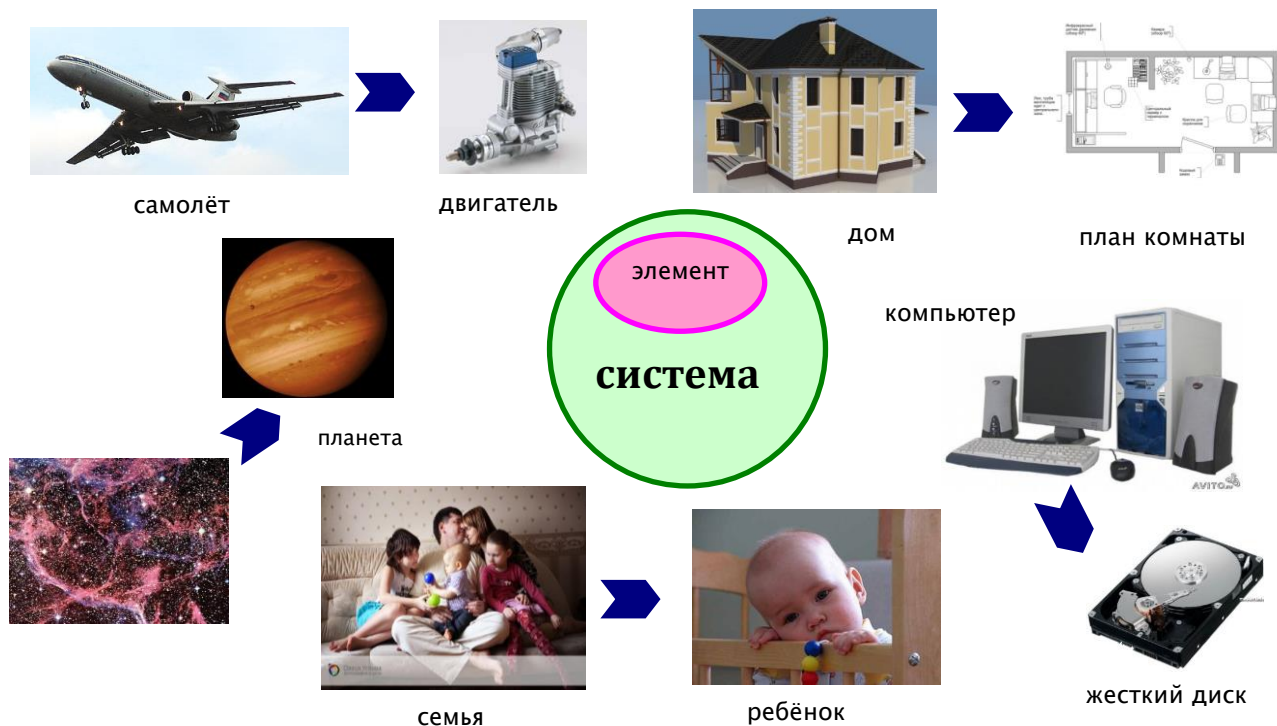
В процессе познания окружающего мира человечество постоянно использует *моделирование* и *формализацию*. При изучении нового объекта сначала обычно строится его описательная информационная модель на естественном языке, затем она формализуется, то есть выражается с использованием формальных языков (математики, логики и др.).

Визуализация формальных моделей. В процессе исследования формальных моделей часто производится их визуализация. Для визуализации алгоритмов используются блок-схемы: пространственных соотношений между объектами – *чертежи*, моделей электрических цепей – *электрические схемы*, логических моделей устройств – *логические схемы* и так далее.

Визуальные модели обычно являются интерактивными, то есть исследователь может менять начальные условия и параметры протекания процессов и наблюдать изменения в поведении модели.

Системный подход в моделировании. Практически каждый объект состоит из других объектов, то есть представляет собой **систему**. Важным признаком системы является ее *целостное функционирование*. Система является не набором отдельных элементов, а совокупностью взаимосвязанных элементов. *Например*, компьютер является системой, состоящей из различных устройств, при этом устройства связаны между собой и аппаратно (физически

подключены друг к другу) и функционально (между устройствами происходит обмен информацией).



Система является совокупностью взаимосвязанных объектов, которые называются элементами системы.

Состояние системы характеризуется ее **структурой**, то есть составом и свойствами элементов, их отношениями и связями между собой. Система сохраняет свою целостность под воздействием различных внешних воздействий и внутренних изменений до тех пор, пока она сохраняет неизменной свою структуру. Если структура системы меняется (*например*, удаляется один из элементов), то система может перестать функционировать как целое. Так если удалить одно из устройств компьютера (*например*, процессор), компьютер выйдет из строя, то есть прекратит свое существование как система.

Статические информационные модели. Любая система существует в пространстве и во времени. В каждый момент времени система находится в определенном состоянии, которое характеризуется составом элементов, значениями их свойств, величиной и характером взаимодействия между элементами и так далее.

Модели, описывающие состояние системы в определенный момент времени, называют **статическими информационными моделями**.

В физике примером статических информационных моделей являются модели, описывающие простые механизмы, в биологии – модели строения растений и животных, в химии – модели строения молекул и кристаллических решеток.

Динамические информационные модели. Состояние системы изменяется во времени, то есть происходят процессы **изменения** и **развития систем**.

Модели, описывающие процессы изменения и развития систем, называются **динамическими информационными моделями**.

В физике динамические информационные модели описывают движение тел, в биологии – развитие организмов или популяций животных, в химии – процессы происхождения химических реакций.

3. Типы информационных моделей.

Табличные информационные модели. Одним из наиболее часто используемых типов информационных моделей является прямоугольная **таблица**, которая состоит из строк и столбцов. Такой тип моделей применяется для описания ряда объектов, обладающих одинаковыми наборами свойств. С помощью таблиц могут быть построены как статические, так и динамические информационные модели в различных предметных областях. Широко известно табличное представление математических функций, статистических данных и так далее.

В **табличной информационной модели** обычно перечень объектов размещен в ячейках первого столбца таблицы, а значения их свойств – в других столбцах. Иногда используется другой вариант размещения данных в табличной модели, когда перечень объектов размещается в первой строке таблицы, а значения их свойств – в последующих строках. Подобным образом организованы таблицы истинности логических функций, рассмотренные в лекции 2. Перечень логических переменных и функций размещен в первой строке таблицы, а их значения – в последующих строках.

В **табличной информационной модели** перечень однотипных объектов или свойств размещен в первом столбце (или строке) таблицы, а значения их свойств размещаются в следующих столбцах (или строках) таблицы.

Представление объектов и их свойств в форме таблицы часто используется в научных исследованиях. Так, на развитие химии и физики решающее влияние оказало создание Д.И. Менделеевым периодической системы элементов, которая представляет собой табличную информационную модель. В этой модели химические элементы располагаются в ячейках таблицы по возрастанию атомных весов, а в столбцах – по количеству валентных электронов, причем по положению в таблице можно определить некоторые физические и химические свойства элементов.

Иерархические информационные модели. Каждый объект обладает определенными свойствами. Однако некоторые группы объектов имеют одинаковые общие свойства, которые отличают их от объектов других групп.

Группа объектов, обладающих одинаковыми общими свойствами, называется *классом объектов*. Внутри класса объектов могут быть выделены подклассы, объекты которых обладают некоторыми особенными свойствами, в свою очередь подклассы могут делиться на еще более мелкие группы и так далее. Такой процесс систематизации объектов называется *процессом классификации*.

В процессе классификации объектов часто строятся информационные модели, которые имеют **иерархическую структуру**. В биологии весь животный мир рассматривается как иерархическая система (тип, класс, отряд, семейство, род, вид), в информатике используется иерархическая файловая система и т.д.

Статическая иерархическая модель. Рассмотрим процесс построения информационной модели, которая позволяет классифицировать современные компьютеры. Класс **Компьютеры** можно разделить на три подкласса: *Суперкомпьютеры*, *Серверы* и *Персональные компьютеры*.

Компьютеры, входящие в подкласс *Суперкомпьютеры*, отличаются сверхвысокой производительностью и надежностью и используются в крупных научно-технических центрах для управления процессами в реальном масштабе времени.

Компьютеры, входящие в подкласс *Серверы*, обладают высокой производительностью и надежностью и используются в качестве серверов в локальных и глобальных сетях.

Компьютеры, входящие в подкласс *Персональные компьютеры*, обладают средней производительностью и надежностью и используются в офисах и дома для работы с различными приложениями.

Подкласс *Персональные компьютеры* делится, в свою очередь, на *Настольные*, *Портативные* и *Карманные* компьютеры.

В иерархической структуре элементы распределяются по уровням, от первого (верхнего) уровня до нижнего (последнего) уровня. На первом уровне может располагаться только один элемент, который является «вершиной» иерархической структуры. Основное отношение между уровнями состоит в том, что элемент более высокого уровня может состоять из нескольких элементов нижнего уровня, при этом каждый элемент нижнего уровня может входить в состав только одного элемента верхнего уровня.

В **иерархической информационной модели** объекты распределены по уровням. Каждый элемент более высокого уровня может состоять из элементов нижнего уровня, а элемент нижнего уровня может входить в состав только одного элемента более высокого уровня.

В рассмотренной иерархической модели, классифицирующей компьютеры, имеются три уровня. На первом, верхнем, уровне располагается элемент *Компьютеры*, в него входят три элемента второго уровня *Суперкомпьютеры*, *Серверы* и *Персональные компьютеры*. В состав последнего входят три элемента третьего, нижнего, уровня *Настольные*, *Портативные* и *Карманные* компьютеры.

Изображение информационной модели в форме графа. *Граф* является удобным способом наглядного представления структуры информационных моделей. *Вершины графа* (овалы) отображают элемент системы.

Элементы верхнего уровня находятся в отношении «состоять из» к элементам более низкого уровня. Такая связь между элементами отображается в форме *дуги графа* (направленной линии в форме стрелки). Графы, в которых связи между объектами несимметричны (как в данном случае), называются *ориентированными*.

Иерархическая модель, классифицирующая компьютеры в виде графа представлена на рис. 1.

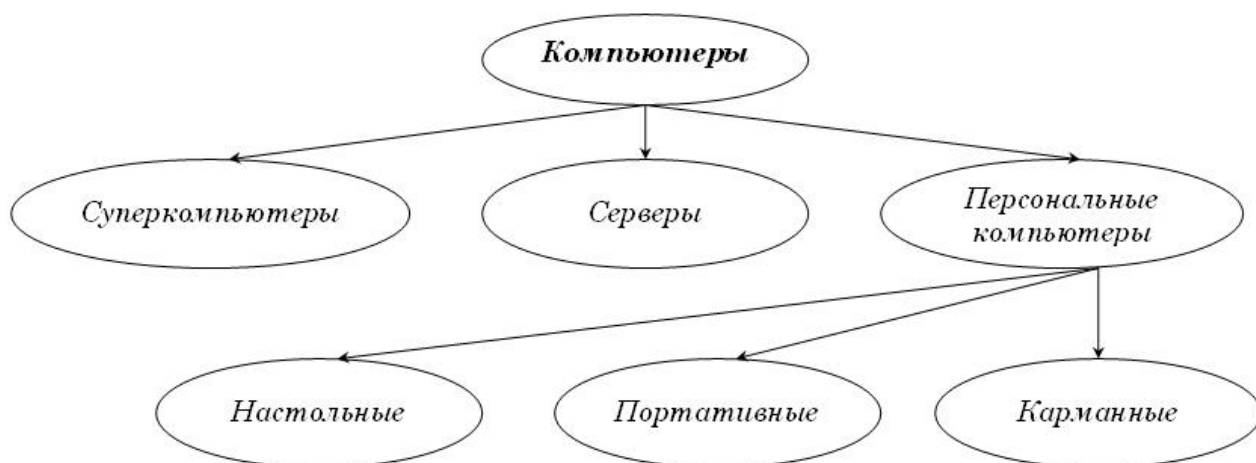


Рис. 1. Классификация компьютеров

Динамическая иерархическая модель. Для описания исторического процесса смены поколений семьи используются динамические информационные модели в форме генеалогического дерева. В качестве примера можно рассмотреть фрагмент генеалогического дерева династии Рюриковичей.

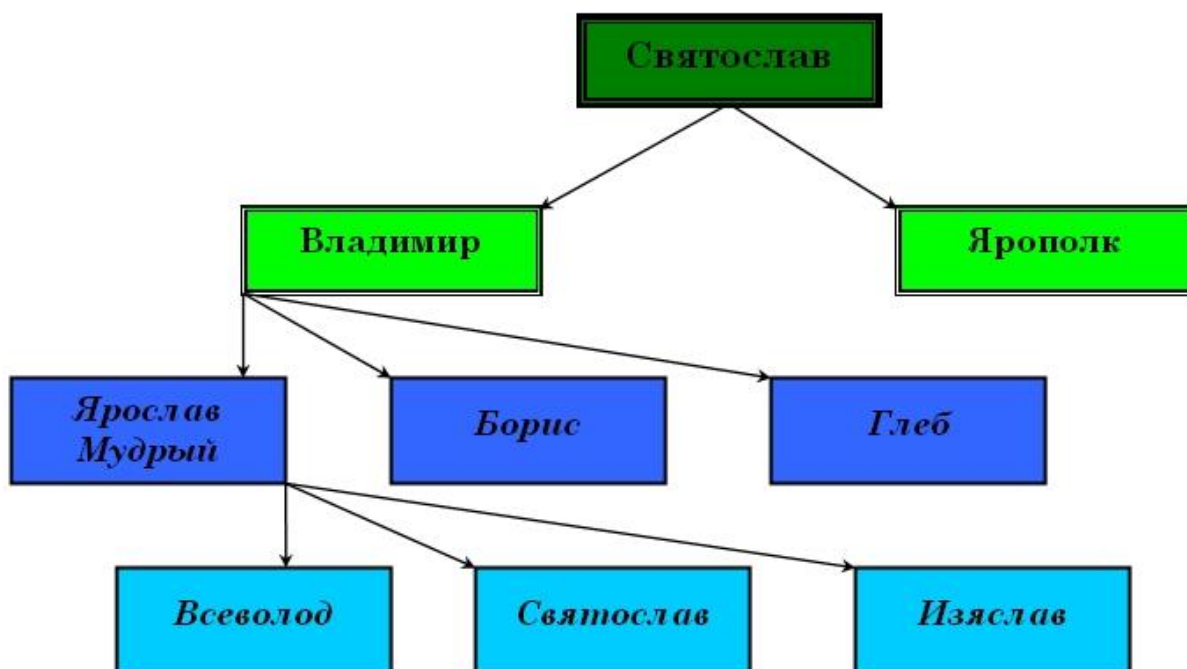


Рис. 2. Генеалогическое дерево Рюриковичей (X – XI века)

Сетевые информационные модели применяются для отражения систем со сложной структурой, в которых связи между элементами имеют произвольный характер.

Например, различные региональные части глобальной компьютерной сети Интернет (американская, европейская, российская, австралийская и т.д.) связаны между собой высокоскоростными линиями связи. При этом одни части (*например*, американская) имеют прямые связи со всеми региональными частями Интернета, а другие могут обмениваться информацией между собой только через американскую часть (*например*, российская и австралийская).

На рис. 3 представлен граф, который отражает структуру глобальной сети Интернет.

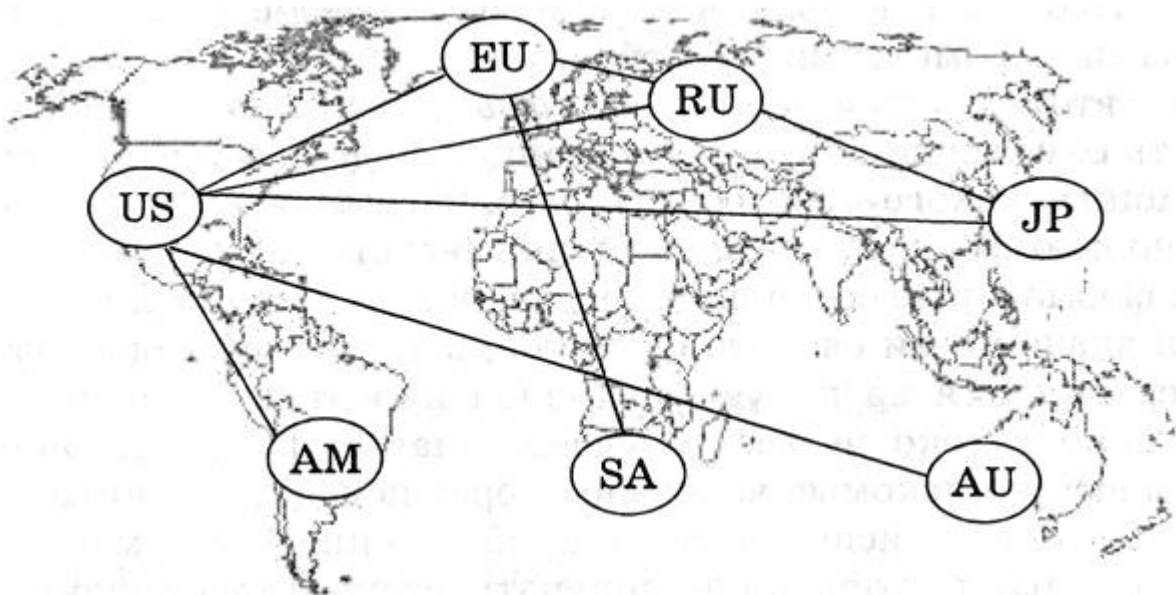


Рис. 10.3. Сетевая структура глобальной сети Интернет

Вершинами графа являются региональные сети. Связи между вершинами носят двусторонний характер и поэтому изображаются ненаправленными линиями (*ребрами*), а сам граф поэтому называется *неориентированным*.

Представленная сетевая информационная модель является статической моделью. С помощью сетевой динамической модели можно, *например*, описать процесс передачи мяча между игроками в коллективной игре (футболе, баскетболе, волейболе).

Контрольные вопросы

1. Что называют математической моделью объекта или явления?
2. Почему невозможно точное исследование поведения объектов или явлений?
3. Может ли объект иметь несколько моделей? Приведите пример.
4. Могут ли разные объекты описываться одной и той же моделью? Если да, приведите пример.
5. Какие бывают модели? Приведите примеры материальных и информационных моделей.
6. Что такое формализация? Приведите примеры формальных моделей.
7. В чем разница между статическими и динамическими информационными моделями? Приведите примеры статических и динамических информационных моделей.
8. Какие системы объектов целесообразно и возможно представлять с помощью табличных моделей?
9. Какие системы объектов целесообразно и возможно представлять с помощью иерархических моделей?
10. Какие системы объектов целесообразно и возможно представлять с помощью сетевых моделей?

Лекция 9.

КОМПЬЮТЕРНЫЕ СЕТИ

1. Основные понятия вычислительной сети.
2. Топология вычислительной сети.
3. Сетевые стандарты.
4. Каналы связи.
5. Передача данных в сети.
6. Коммуникационное оборудование.

1. Основные понятия вычислительной сети.

Компьютерная сеть (англ. *Computer NetWork*, от *net* – сеть и *work* – работа) – совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети.



Абоненты сети – объекты, генерирующие или потребляющие информацию в сети – могут быть отдельные компьютеры, комплексы ЭВМ, промышленные роботы, станки с числовым программным управлением и т.д.

Вычислительные сети подразделяются на два вида:

- **локальные**

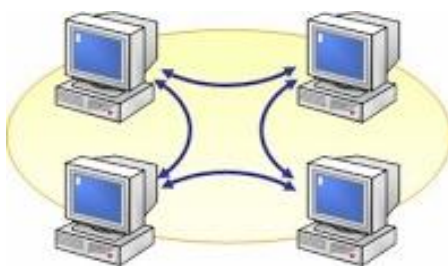
– *глобальные*



Локальные вычислительные сети

(LAN – Local Area

Network) представляют собой совокупность однородных технических систем, объединенных каналами связи и обеспечивающих на небольшой территории (до нескольких км) пользователям доступ к распределенным вычислительным и информационным ресурсам.



Глобальные вычислительные сети

(WAN – Wide Area

Network) – обеспечивают взаимодействие ЛВС на большой территории, расстояние между ЛВС могут достигать десятки тысяч км. Глобальные вычислительные сети могут быть объединены между собой с помощью Интернета.



Разработка любого сетевого проекта начинается с выбора архитектуры будущей компьютерной сети.

Под **архитектурой вычислительной сети** принято понимать объединение топологий сети, определяющих ее конфигурацию, а также сетевых протоколов и стандартов, обеспечивающих ее работоспособность.

В сетях используется технология **«клиент – сервер»**, обеспечивающая распределенную обработку данных. Основной принцип технологии заключает-

ся в разделении функций между сервером и клиентом. Сервер организует доступ к ресурсам общего пользования, а клиент использует ресурсы.

Компьютер, управляющий тем или иным ресурсом, принято называть **сервером** этого ресурса (файл-сервер, сервер приложений, сервер базы данных), а компьютер, имеющий возможность использовать ресурс, – **клиентом**.

Программное обеспечение, обслуживающее клиент-серверную технологию, подразделяется на программное обеспечение, реализующее функции сервера (оно устанавливается на компьютере-сервере), и программное обеспечение, обеспечивающее доступ клиентов к ресурсам сети (оно устанавливается на компьютерах-клиентах).

К основным характеристикам сетей относятся:

Пропускная способность – максимальный объем данных, передаваемых сетью в единицу времени. Пропускная способность измеряется в Мбит/с.

Время реакции сети – время, затрачиваемое программным обеспечением и устройствами сети на подготовку к передаче информации по данному каналу. Время реакции сети измеряется миллисекундах.

2. Топология вычислительной сети.

Топология вычислительной сети – определяет ее логическую структуру, т.е. маршруты передачи данных между узлами сети, требования к соответствующему коммутационному оборудованию и его настройке. Существует три базовые топологии: «**кольцо**», «**общая шина**», «**звезда**».

Шинная структура – соединение компьютеров производится посредством одного кабеля (все компьютеры подключены к единому каналу). Данные от компьютера передаются всем компьютерам сети, однако воспринимаются только тем компьютером, адрес которого указан в передаваемом сообщении. При этом в каждый момент времени только один компьютер может вести передачу. К недостаткам шинной топологии сети относится уменьшение пропускной способности сети при значительных объемах передаваемой информации. Преимуществом шинной топологии является гибкость сети, выражающаяся в том, что подключение и отключение машин не требует прерывания работы сети.

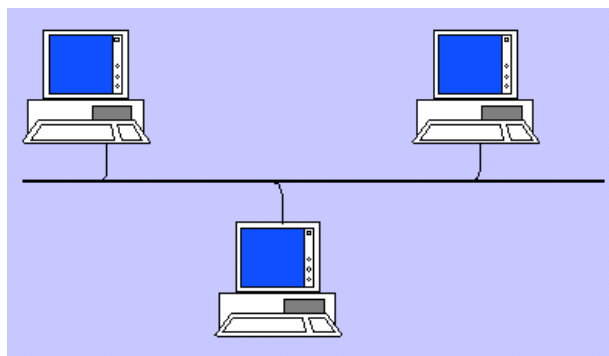


Рис. 1. Шинная структура

Кольцевая структура предусматривает объединение всех компьютеров с помощью кабельной системы, имеющей форму замкнутого круга. Данные передаются от одного компьютера к другому; при этом если один компьютер получает данные, предназначенные для другого компьютера, то он передает их дальше (по кольцу).

К недостаткам кольцевых сетей следует отнести малую гибкость архитектуры. Выход из строя любого из элементов сети приводит к потере ее работоспособности.

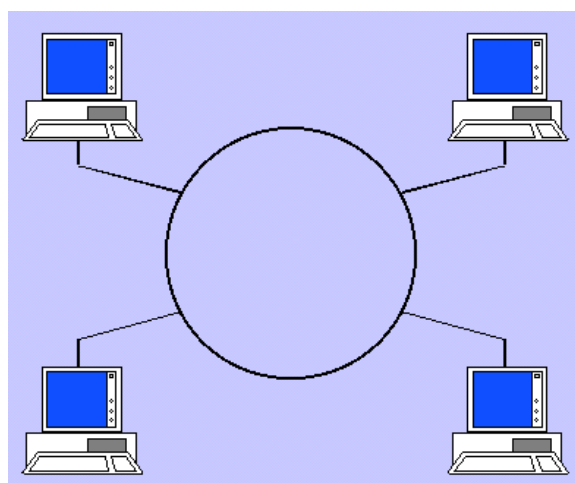


Рис. 2. Кольцевая структура

Звездная структура предполагает наличие центрального компьютера, с которым связаны все остальные ПК. Основным преимуществом такой топологии является ее устойчивость к сбоям, возникающим вследствие неполадок на отдельных ПК или из-за повреждения сетевого кабеля, а также высокая скорость обмена информацией в вычислительной сети.

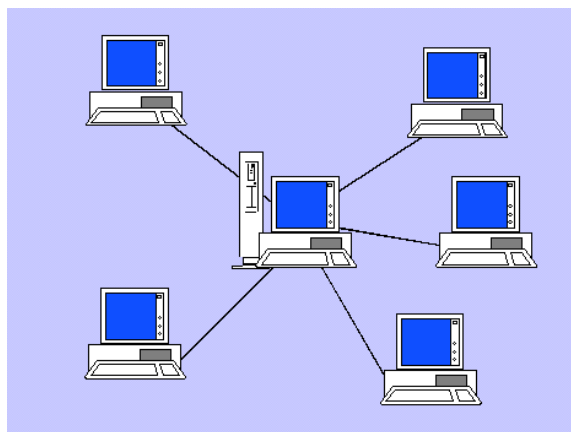


Рис. 3. Звездная структура

В настоящее время используется **полносвязная топология**, комбинирующая базовые топологии: звезда – шина, звезда – кольцо, при которой каждый компьютер непосредственно соединен со всеми компьютерами вычислительной сети.

Разнообразие топологий локальных вычислительных сетей и методов доступа к ним породило соответствующие разновидности протоколов канального и физического уровней управления.

3. Сетевые стандарты.

Концептуальной основой построения вычислительных сетей является базовая эталонная модель взаимосвязи открытых систем **OSI** (*Open Systems Interconnection*), которая разработана международной организацией по стандартизации **ISO** (*International Standards Organization*).

Модель OSI включает семь уровней управления, обеспечивающих взаимодействие прикладных процессов пользователей и программ.

Уровень 1 – **физический** – реализует управление каналом связи, что сводится к подключению и отключению каналов и формированию сигналов, представляющих передаваемые данные.

Уровень 2 – **канальный** – обеспечивает надежность передачи данных, используя средства контроля и обнаружения ошибок.

Уровень 3 – **сетевой** – выбирает маршрут передачи данных по линиям связи.

Уровень 4 – **транспортный** – обеспечивает обмен данными (сопряжения) между сетью передачи данных и абонентами сети.

Уровень 5 – **сеансовый** – организует сеансы связи на период взаимодействия процессов пользователей.

Уровень 6 – **представления** – обеспечивает взаимодействие разнородных систем через преобразование различных форматов данных и кодов.

Уровень 7 – **прикладной** – реализует сетевые вычислительные услуги (почта, телеконференция и т.п.).

Как правило, уровни 1 и 2 реализуются техническими средствами, а уровни 3 – 7 – программными средствами.

Процедура взаимодействия между процессами одного уровня управления с другими уровнями на основе обмена сообщениями называется *протоколом*.

В Интернете на транспортном уровне используется протокол **TCP** (*Transmission Control Protocol*), а на сетевом уровне – протокол **IP** (*Internet Protocol*). Эти два протокола взаимосвязаны, поэтому их часто объединяют и называют протоколом **TCP/IP**.

Протокол **TCP** реализует на узле отправителя пакетирование сообщений (разбивку сообщения на части), а на узле-получателе – сборку пакетов в единое сообщение.

Протокол **IP** обеспечивает передачу информации по сети, определяет маршрут передачи информации и формат представления адресов.

4. Каналы связи.

Средой передачи информации называются те линии связи (или каналы связи), по которым производится обмен информацией между компьютерами.

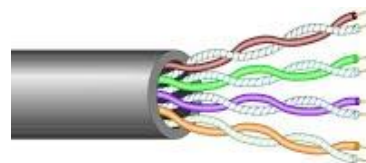
Обмен информацией в вычислительных сетях осуществляется по **проводным** и **беспроводным** каналам связи. Каналы связи характеризуются *пропускной способностью, помехоустойчивостью и стоимостью*.

Проводные каналы связи используют три типа кабелей: **витую пару, коаксиальный и оптоволоконный кабель**. Проводные каналы чаще всего используются для создания локальных компьютерных сетей.

Каждый тип кабеля имеет свои преимущества и недостатки, так что при выборе типа кабеля надо учитывать как особенности решаемой задачи, так и особенности конкретной сети, в том числе и используемую топологию.

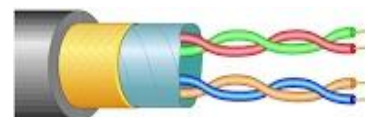
Кабель на основе витых пар представляет собой несколько пар скрученных изолированных медных проводов в единой диэлектрической (пластиковой) оболочке. Он довольно гибкий и удобный для прокладки.

Неэкранированные витые пары характеризуются слабой защищенностью от внешних электромагнитных помех, а также слабой защищенностью от подслушивания с целью, например, промышленного шпионажа.



Перехват передаваемой информации возможен как с помощью контактного метода (посредством двух иголок, воткнутых в кабель), так и с помощью бесконтактного метода, сводящегося к радиоперехвату излучаемых кабелем электромагнитных полей. Для устранения этих недостатков применяется экранирование.

В случае *экранированной витой пары* каждая из витых пар помещается в металлическую оплетку-экран для уменьшения излучений кабеля, защиты от внешних электромагнитных помех и снижения взаимного влияния пар проводов друг на друга.



Коаксиальный кабель представляет собой электрический кабель, состоящий из центрального провода и металлической оплетки, разделенных между собой слоем диэлектрика (внутренней изоляции) и помещенных в общую внешнюю оболочку.



Оптоволоконный (волоконно-оптический) кабель – это принципиально иной тип кабеля. Информация по нему передается не электрическим сигналом, а световым. Главный его элемент – это прозрачное стекловолокно, по которому свет проходит на огромные расстояния с незначительным ослаблением.

Структура оптоволоконного кабеля очень проста и похожа на структуру коаксиального электрического кабеля, только вместо центрального медного провода здесь используется тонкое (диаметром порядка 1-10 мкм) стекловолокно, а вместо внутренней изоляции - стеклянная или пластиковая оболочка, не позволяющая свету выходить за пределы стекловолокна.



К **беспроводным каналам связи** относится радиосвязь и спутниковая

СВЯЗЬ.

В настоящее время широкое распространение получает **WI-FI**-технология, которая позволяет организовывать скоростной доступ в Интернет, без использования проводов.



5. Передача данных в сети.

Имеются три способа передачи информации в вычислительных сетях: *коммутация каналов*, *коммутация сообщений* и *коммутация пакетов*.

При **коммутации каналов** образуется непосредственно физическое соединение двух узлов. Установленное физическое соединение находится в монопольном владении абонентов, независимо от того, используется оно или нет. Примерами глобальных сетей с коммутацией каналов являются аналоговые телефонные сети общего назначения и цифровая сеть ISDN.

При **коммутации сообщений** информация передается порциями в виде готовых документов. На время передачи сообщения канал становится недоступным для других пользователей. Этот вид коммутации используется в системах электронной почты.

При **коммутации пакетов** обмен производится короткими порциями фиксированной структуры и длины. Пакет – это часть сообщения. Малая длина пакета предотвращает блокировку каналов связи и увеличивает их пропускную способность. Пакеты на узле отправителя нумеруются, снабжаются адресами и последовательно передаются по сети по разным, свободным в конкретный момент времени, каналам связи. Пакеты приходят на узел-получатель в произ-

вольном порядке, в зависимости от выбранного в момент отправки канала. Полное сообщение собирается на узле коммутации получателя. Коммутация пакетов повышает эффективность использования каналов связи за счет перераспределения трафика.

Трафиком называется объем информации или поток информации, передаваемой по сети.

6. Коммуникационное оборудование.

Передачу информации между компьютерами в сети обеспечивает коммуникационная система, которая может включать в себя модемы, повторители, коммутаторы, маршрутизаторы и другие устройства.

Модем (акроним, составленный из слов **модулятор** и **демодулятор**) – устройство, применяющееся в системах связи для преобразования цифровой информации в аналоговую, путем модуляции и демодуляции дискретных и аналоговых электрических сигналов. Модемы широко применяются для связи компьютеров через телефонную сеть (*телефонный модем*), кабельную сеть (*кабельный модем*), радиоволны (радиорелейная связь).

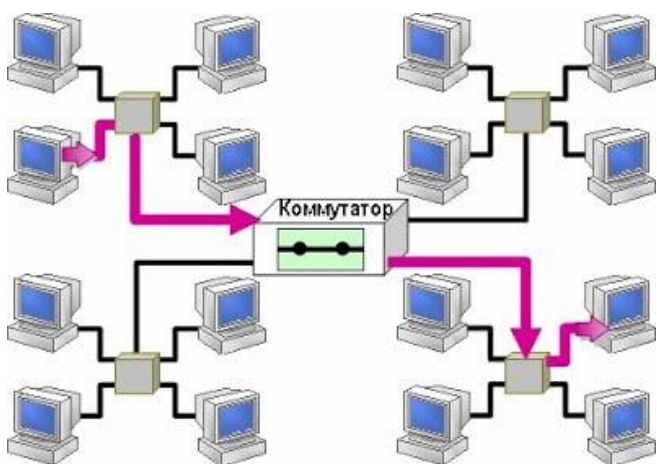


Повторитель – устройство, обеспечивающее усиление сигнала без изменения его информативности. По мере передвижения по линиям связи сигналы затухают. Для уменьшения влияния затухания используются повторители. Повторитель не только усиливает принимаемые сигналы, но и восстанавливает их характеристики, уменьшая помехи. Повторитель работает на физическом уровне модели OSI.



Мультиплексор – устройство или программа, позволяющие передавать по одной коммуникационной линии одновременно несколько различных потоков данных за счет частотного разделения канала.

Концентратор – сетевое устройство для объединения нескольких устройств локальной сети в общий сегмент сети. Концентратор работает на физическом уровне модели OSI. В последнее время концентраторы используются довольно редко, вместо них получили распространение коммутаторы.



Коммутаторы – устройства, работающие на канальном уровне модели OSI и повышающие производительность сети путем логического выделения каждого подключенного устройства в отдельный сегмент.

Мост – устройство, соединяющее локальные сети разных топологий, работающих под управлением однотипных сетевых операционных систем, имеющих одинаковые протоколы. Сетевой мост работает на втором уровне модели OSI.

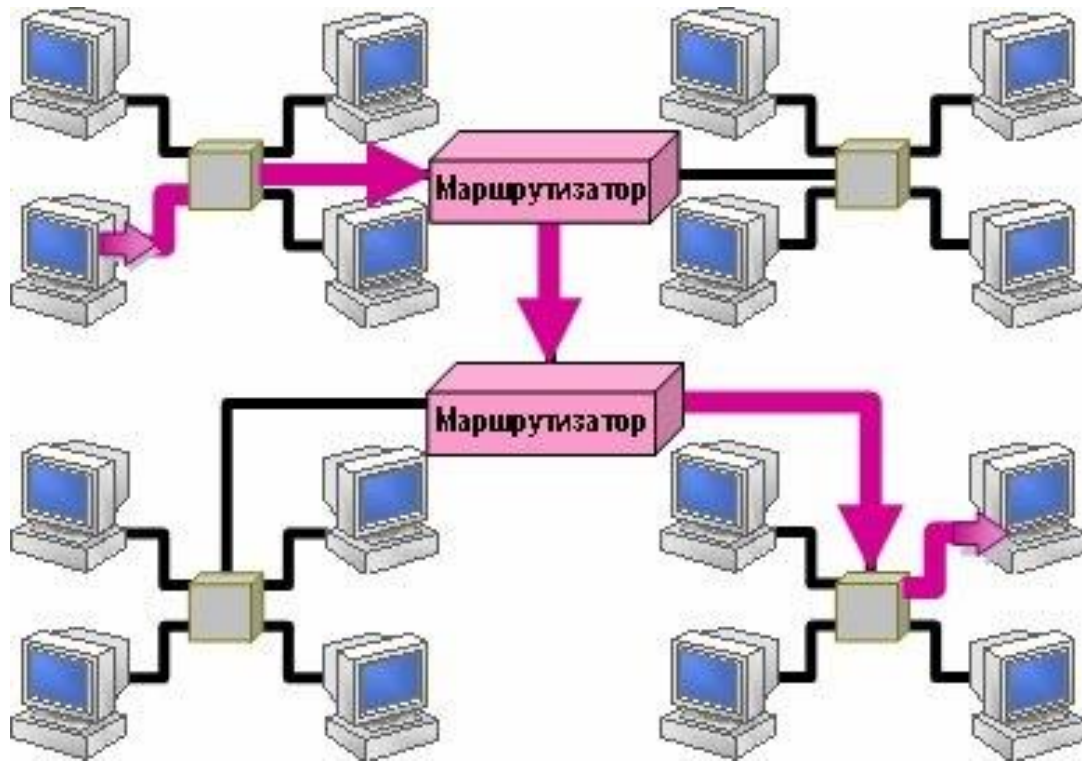
Определение пути, по которому будет передано сообщение или пакет, осуществляется с помощью специального устройства – **маршрутизатора** (или роутера). Протокол IP выполняет функции маршрутизации (третий уровень модели OSI), выбирая путь для передачи сообщений. На каждом узле маршрутизатор определяет, в каком направлении передать поступившее сообщение.

Для поиска лучшего маршрута к любому адресату в сети используются таблицы маршрутизации.

Интернет объединяет сети, которые работают по различным правилам. Для согласования этих правил (протоколов) служат специальные устройства, называемые **плюзами**.



С помощью шлюзов можно подключить локальную вычислительную сеть к глобальной вычислительной сети.



Контрольные вопросы

1. Что такое компьютерная сеть и что необходимо для создания компьютерных сетей?
2. Какова основная задача, решаемая при создании компьютерных сетей?
3. Как следует рассматривать архитектуру компьютерных сетей согласно модели ISO/OSI?
4. Что такое протоколы? Для чего они предназначены?
5. Что такое шлюзы? Какими могут быть шлюзы?
6. Что такое серверы сети?
7. Каковы основные достоинства и недостатки локальных сетей:
 - одноранговых с топологией линейная шина;
 - одноранговых с топологией звезда;
 - сетей на основе сервера?
9. Какие существуют способы передачи информации в вычислительных сетях?
10. Каковы основные различия между концентратором, коммутатором и маршрутизатором?

ГЛОССАРИЙ

А

Адаптер – устройство связи компьютера с периферийными устройствами.

Адекватность информации – уровень соответствия создаваемого с помощью полученной информации образа реальному объекту, процессу, явлению и т.п.

Адрес в Интернете – полный путь к ресурсу в Интернете, включающий в себя протокол для доступа к ресурсам, идентификатор сервера и путь к ресурсу на этом сервере.

Адрес электронной почты – идентификатор конечного пользователя, определяющий почтовый ящик на конкретном сервере.

Алгоритм – это четкое описание строго определенной последовательности действий, которые необходимо выполнить для преобразования исходных данных в результат решения задачи.

Алгоритмизация – процесс разработки алгоритма (плана действий) для решения конкретной задачи.

Алфавит – это все множество символов, используемых в некотором языке для представления информации.

Антивирусная программа – программа, предотвращающая заражение компьютерным вирусом и ликвидирующая последствия заражения.

Аппаратное обеспечение персонального компьютера – система взаимосвязанных технических устройств, выполняющих ввод, хранение, обработку и вывод информации.

Архитектура фон Неймана – архитектура компьютера, имеющего одно арифметико-логическое устройство, через которое проходит поток данных, и одно устройство управления, через которое проходит поток команд.

Архитектура ЭВМ – концептуальная структура вычислительной машины, определяющая проведение обработки информации и включающая методы преобразования информации в данные и принципы взаимодействия технических средств и программного обеспечения.

Аудиоадаптер (звуковая плата) – специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Б

Байт – группа из восьми битов информации, основная единица количества информации.

Бит – единица информации, содержащееся в сообщении, уменьшающее неопределенность знаний в два раза; двоичная единица измерения количества информации, хранящейся в компьютере.

Брандмауэр – межсетевой экран, который отслеживает передачу данных между сетями, осуществляет контроль текущих соединений, выявляет подозрительные действия и тем самым предотвращает несанкционированный доступ из Интернета в локальную сеть.

Браузер – программа-клиент, облегчающая процессы перемещения между узлами

глобальной сети, отображения, поиска, сбора и хранения информации.

Буфер обмена – область памяти, которая служит для временного хранения данных, предназначенных для обмена.

В

Видеоадаптер – специальная плата ПК, обеспечивающая формирование изображения на экране монитора от информации, передаваемой процессором.

Видеопамять – часть оперативной памяти, отведенная для хранения данных, которые используются для формирования изображения на экране монитора.

Витая пара – вид кабеля связи, представляет собой одну или несколько пар изолированных проводников, скрученных между собой для уменьшения взаимных наводок при передаче сигнала.

Внешняя память – совокупность запоминающих устройств для длительного хранения данных.

Внутренняя память – запоминающее устройство, напрямую связанное с процессором и предназначенное для хранения выполняемых программ и данных, непосредственно участвующих в вычислениях.

Г

Глобальная сеть – множество объединенных сетей, в котором для межсетевых коммуникаций используются сторонние средства коммуникации.

Графическая операционная система – операционная система, в которой основным средством управления является мышь или другой аналогичный манипулятор.

Графический редактор – программа или комплекс программ, позволяющих создавать и редактировать изображения на экране компьютера: рисовать линии, раскрашивать области экрана, создавать надписи различными шрифтами, обрабатывать изображения, полученные с помощью сканеров.

Графопостроитель – устройство для вывода информации в виде графиков и чертежей на неподвижную или вращающуюся на барабане бумагу.

Д

Данные – это информация, представленная в формализованном виде и предназначенная для обработки ее техническими средствами.

Двоичное кодирование – кодирование информации в виде 0 и 1.

Двоичный код – это способ представления данных в виде комбинации двух знаков, обычно обозначаемых цифрами 0 и 1.

Декодирование – является обратным к кодированию процессом.

Джойстик – стержень-ручка, отклонение которой от вертикального положения приводит к передвижению курсора в соответствующем направлении по экрану монитора. Часто применяется в компьютерных играх.

Диаграмма – это средство наглядного графического изображения информации, предназначенное для сравнения нескольких величин или нескольких значений одной величины, слежения за изменением их значений и т.п.

Диск – круглая металлическая или пластмассовая пластина, покрытая магнитным материалом, на которую информация наносится в виде концентрических дорожек, разделённых на секторы.

Дисковод – устройство, управляющее вращением магнитного диска, чтением и записью данных на нём.

Дискретизация – это преобразование непрерывных изображений и звука в набор дискретных значений в форме кодов.

Дискретный сигнал – сигнал, если его параметр в заданных пределах может принимать отдельные фиксированные значения.

Дисплей – устройство визуального отображения информации.

Дистрибутив – программа, предназначенная для начальной установки приложений.

Длина кода – количество знаков, используемых для представления кодируемой информации.

Документ – носитель информации, подтверждающий какой-либо факт или событие.

Драйвер – это специальная вспомогательная программа, управляющая внешними устройствами компьютера или управляющая выполнением программ.

З

Защита информации – мероприятия, направленные на обеспечение информационной безопасности.

Звуковая карта – дополнительный элемент компьютера, не относящийся к его основному назначению, позволяющий обрабатывать звук.

Знак – это элемент некоторого конечного множества отличных друг от друга сущностей.

И

Идентификация – процесс отождествления объекта (сети, пользователя, системы, канала, устройства, программы, процесса) с одним из известных.

Инструментальные программные средства – программы, используемые в ходе разработки, корректировки или развития других программ: редакторы, отладчики, вспомогательные системные программы, графические пакеты и др. По назначению близки системам программирования.

Интернет – глобальная компьютерная сеть, объединяющая сети, шлюзы, серверы и компьютеры, использующие для связи единый набор протоколов.

Интернет-провайдер – организация, которая предоставляет возможность подключения к сети Интернет и доступ к различным его сервисам.

Интерпретатор – разновидность транслятора. Переводит и выполняет программу с языка высокого уровня в машинный код строка за строкой.

Интерфейс – это совокупность унифицированных стандартных соглашений, аппаратных и программных средств, методов и правил взаимодействия устройств или программ, а также устройств или программ с пользователем.

Информатизация – это сложный социальный процесс, связанный со значительными изменениями в образе жизни населения. Он требует серьёзных усилий на многих направлениях, включая ликвидацию компьютерной неграмотности, формирование культуры использования новых информационных технологий и др.

Информатизация общества – организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения

информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления организаций, общественных объединений на основе формирования и использования информационных ресурсов.

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

Информационная безопасность – это защищенность информации от случайных и намеренных действий, способных нанести недопустимый ущерб.

Информационная культура общества предполагает умение целенаправленно работать с информацией и использовать ее для получения, обработки и передачи в компьютерную информационную технологию.

Информационная модель – модель объекта, представленная в виде информации, описывающей существенные для данного рассмотрения параметры и переменные величины объекта, связи между ними, входы и выходы объекта.

Информационная система осуществляет сбор, передачу и переработку информации об объекте, снабжающую работников различного уровня информацией для реализации функции управления.

Информационная технология – это совокупность научных методов и технических приемов производства информационных продуктов и услуг с применением всего многообразия средств вычислительной техники и связи.

Информационная услуга – услуга, ориентированная на удовлетворение информационных потребностей пользователей путем предоставления информационных продуктов.

Информационно-поисковая система – система, выполняющая функции хранения большого объема информации, быстрого поиска требуемой информации, добавления, удаления и изменения хранимой информации, вывода её в удобном для человека виде.

Информационный подход – метод научного познания объектов, процессов или явлений природы и общества, согласно которому в первую очередь выявляются и анализируются наиболее характерные информационные аспекты, определяющие функционирование и развитие изучаемых объектов.

Информационный процесс – это совокупность последовательных действий, производимых над информацией с целью получения результата.

Информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности и неполноты знаний.

Инфракрасная мышь – устройство управления манипуляторного типа, отличается от обычной мыши наличием устройства беспроводной связи с системным блоком.

К

Канал связи – система технических средств и среда распространения сигналов для передачи сообщений (не только данных) от источника к получателю (и наоборот).

Каталог (директория, папка) – оглавление файлов. Доступен пользователю через командный язык операционной системы. Его можно просматривать, переименовывать зарегистрированные в нём файлы, переносить их содержимое на новое место и удалять. Часто имеет иерархическую структуру.

Качество информации – совокупность свойств, отражающих степень пригодности конкретной информации об объектах и их взаимосвязях для достижения целей, стоящих перед пользователем.

Классификация – распределение объектов на классы и подклассы на основании общих признаков.

Кластер – минимальная единица размещения информации на диске, состоящая из одного или нескольких смежных секторов, которая может быть отведена файлу.

Клиент (рабочая станция) – любой компьютер, имеющий доступ к услугам сервера. Клиентом также называют прикладную программу, которая от имени пользователя получает услуги сервера.

Код – система условных обозначений или сигналов.

Кодирование информации – это процесс формирования определенного представления информации. При кодировании информация представляется в виде дискретных данных.

Кодировка символов – соответствие между набором букв и числами.

Количество информации – мера уменьшения неопределенности знаний.

Команда – описание элементарной операции, которую должен выполнить компьютер. Обычно содержит код выполняемой операции, указания по определению операндов (или их адресов), указания по размещению получаемого результата.

Коммутатор – сетевое устройство для соединения компьютеров в пределах одного сегмента локальной сети. В отличие от концентратора, который распространяет пакеты от одного компьютера ко всем остальным, коммутатор передает данные только компьютеру-получателю.

Компакт-диск (CD) – оптический носитель информации в виде пластикового диска с отверстием в центре, процесс записи и считывания информации которого осуществляется при помощи лазера.

Компилятор – разновидность транслятора. Читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Компьютер – устройство для приема, переработки, хранения и выдачи информации в электронном виде, которая может воспринимать и выполнять сложные последовательности вычислительных операций по заданной инструкции – программе.

Компьютеризация – задачи массового внедрения компьютеров во все области жизни, стоящие перед странами как необходимое важное условие их прогресса и развития, а также последствия, которые будут вызваны этим массовым внедрением компьютеров. Цель компьютеризации – улучшение качества жизни людей за счёт увеличения производительности и облегчения условий их труда.

Компьютерная сеть – это совокупность компьютеров и различных устройств, обеспечивающих информационный обмен между компьютерами в сети без использования каких-либо промежуточных носителей.

Компьютерное преступление – это противоправное, виновно совершенное, нака-

зубое в уголовном порядке общественно опасное деяние, причиняющие вред, либо создающее угрозу причинения вреда общественным отношениям по законному использованию компьютерной информации.

Компьютерный вирус – это умышленно созданная программа, способная самопроизвольно присоединяться к другим программам, создавать свои копии, внедрять их в файлы, системные области компьютера и в другие, объединенные с ним компьютеры, с целью нарушения их работы и порчи информации.

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

Концентратор – сетевое устройство для объединения нескольких компьютеров в локальную сеть, повторяет входящий на один порт сигнал на все активные порты.

Курсор – это экранный элемент, указывающий место ввода знаковой информации.

Кэш-память – память с быстрым доступом, предназначенная для временного хранения данных.

Л

Линейный алгоритм – алгоритм, у которого все его действия выполняются последовательно одно за другим.

Локальная сеть – сеть в пределах предприятия, учреждения, одной организации.

М

Манипуляторы (джойстик, мышь, трекбол и др.) – специальные устройства для управления курсором.

Маршрут – последовательность узлов сети, через которые передается информация от источника к приемнику.

Маршрутизатор – сетевое устройство для передачи данных между различными сегментами локальной сети с учетом таблицы маршрутизации.

Маршрутизация – процесс определения в сети пути, по которому блок данных может дойти до адресата. Маршрутизация обеспечивает преобразование адреса объекта назначения в перечень каналов, по которым этот блок данных проследует к адресату.

Материнская плата – самостоятельный элемент компьютера, который управляет внутренними связями и с помощью системы прерываний взаимодействует с внешними устройствами.

Межсетевой экран – аппаратно-программное средство, которое предотвращает несанкционированный доступ в защищаемую сеть.

Меню – список взаимосвязанных команд, используемых для выполнения задач при работе с ОС Windows или приложениями.

Микропроцессор – процессор (устройство, отвечающее за выполнение арифметических, логических операций и операций управления, записанных в машинном коде), реализованный в виде одной микросхемы или комплекта из нескольких специализированных микросхем.

Микрофон – электроакустический прибор, преобразовывающий звуковые колебания в колебания электрического тока.

Модем – устройство, обеспечивающее преобразование цифровых сигналов компьютера в переменный ток частоты звукового диапазона (модуляцию), а также обратное преобразование (демодуляцию). Используется для обмена информацией между удаленными компьютерами по каналам связи.

Монитор – устройство визуального представления данных.

Мост – простейшее устройство для соединения между собой двух локальных сетей, использующих одинаковые протоколы.

Мощность алфавита – число символов в алфавите.

Мультимедиа технология – собирательное понятие для различных компьютерных технологий, при которых используется несколько информационных сред, таких, как графика, текст, видео, фотография, движущиеся образы (анимация), звуковые эффекты, высококачественное звуковое сопровождение.

Мышь – устройство управления манипуляторного типа. Ее движения синхронизированы с перемещением курсора по экрану монитора.

Н

Накопитель на жёстких магнитных дисках (винчестер) – устройство хранения информации, основанное на принципе магнитной записи. Является основным накопителем данных в большинстве компьютеров.

Неграфическая операционная система – операционная система, в которой основным устройством управления является клавиатура. Управление компьютером осуществляется вводом команд в поле командной строки.

Неопределенность – отсутствие или недостаток определения чего-либо.

Непрерывный (аналоговый) способ представления информации – представление информации, в котором сигнал на выходе датчика будет меняться вслед за изменениями соответствующей физической величины.

Непрерывный сигнал – сигнал, если его параметр в заданных пределах может принимать любые промежуточные значения.

Носитель информации – материальный объект, предназначенный для хранения и передачи информации.

Ноутбук – портативный персональный компьютер, в корпусе которого объединены типичные компоненты ПК, включая монитор, клавиатуру и устройство указания (обычно сенсорная панель, или тачпад), карманный компьютер, а также аккумуляторные батареи.

О

Обобщение информации – преобразование информации о простых частных событиях в информацию о наличии некоторого события более высокого уровня, проявляющегося в частных событиях.

Обработка информации – в информатике – любое преобразование информации из одного вида в другой, производимое по строгим формальным правилам.

Объект – элемент информационной системы, сведения о котором хранятся в базе данных.

Одномерные массивы – упорядоченная последовательность информации, элементы которой оформляются в виде столбца или строки. При этом одномерные массивы имеют только один индекс – номер строки I или номер столбца J .

Оперативное запоминающее устройство (ОЗУ) – устройство энергозависимой памяти компьютера на микросхемах для записи, считывания и хранения информации.

Операционная оболочка – надстройка над операционными системами, которая предоставляет пользователю качественно новый интерфейс и освобождает его от детального знания операций и команд операционной системы.

Операционная система – это комплекс программ, обеспечивающих управление работой компьютера и его взаимодействие с пользователем.

Оптоволоконный канал – использует оптоволокно для передачи сигнала, реализуется в виде стеклянной или пластиковой нити, внутри которой благодаря эффекту полного внутреннего отражения распространяется световой сигнал.

П

Пакеты прикладных программ (ППП) – специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

Папка (каталог) – объект Windows, предназначенный для объединения файлов и других папок в группы.

Пароль – конфиденциальный признак, подтверждающий право пользователя, или прикладной программы на использование какого-нибудь ресурса.

Пенмаус – представляет собой аналог шариковой авторучки, на конце которой вместо пишущего узла установлен узел, регистрирующий величину перемещения.

Переменная – величина, значение которой может меняться в процессе выполнения программы.

Периферийное устройство – устройство, которое подключается к аппаратному компоненту в компьютере, с помощью которого он взаимодействует с внешней средой.

Персональный компьютер – универсальная техническая система, рассчитанная на одного пользователя и управляемая одним человеком.

Пиксель – элементарная точка изображения на экране дисплея.

Пиктограмма – компактное графическое представление объекта или действия в виде условного значка.

Плоттер (графопостроитель) – устройство для автоматического вычерчивания с большой точностью рисунков, схем, сложных чертежей, карт и другой графической информации на бумаге размером до А0 или кальке.

Поколения компьютеров – условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с ними.

Политика сети – совокупность приемов разделения и ограничения прав участников компьютерной сети.

Пользователь – человек, использующий компьютер для решения какой-либо прикладной задачи.

Порты – специальные аппаратно-логические устройства, отвечающие за связь процессора с другими устройствами.

Постоянная память (ПЗУ) – энергонезависимое запоминающее устройство, изго-

товленное в виде микросхемы. Используется для хранения данных, не требующих изменения.

Почтовый ящик – область жесткого диска на почтовом сервере, предназначенная для записи писем. Владелец почтового ящика может с помощью специальной программы-клиента просматривать, копировать, перемещать, удалять содержимое ящика.

Прикладная программа – любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

Приложение – одна или несколько взаимосвязанных программ, которые используются для создания компьютерных объектов: текстов, рисунков, программ и т.п.

Принтер – печатающее устройство, предназначенное для вывода текстовой и графической информации на бумагу.

Проводник – служебная программа, относящаяся к категории диспетчеров файлов.

Программное обеспечение – это совокупность программных и документальных средств для создания и эксплуатации систем обработки данных средствами вычислительной техники.

Программные средства (software) – совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ.

Программы – это упорядоченные последовательности команд.

Прокси-сервер – компьютер, который используется для снижения нагрузки в сети и повышения быстродействия.

Пропускная способность – максимальный объем данных, передаваемых сетью в единицу времени.

Протокол коммуникации – согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных.

Р

Рабочая группа – группа сотрудников, работающих над одним проектом в рамках локальной сети.

Рабочая станция – абонентская система, специализированная на решении определенных задач пользователя.

Рабочий стол – это графическая среда, на которой отображаются объекты и элементы управления операционной системы Windows.

Разветвляющийся алгоритм – алгоритм, у которого порядок выполнения шагов изменяется в зависимости от заданных условий.

Разряд числа – позиция, которую занимает цифра.

Разрядность шины – количество одновременно передаваемых по шине бит.

Региональная сеть – сеть, в которой используются технологии глобальных сетей и которая соединяет локальные сети внутри конкретного географического региона.

Регистр – специальная запоминающая ячейка, выполняющая функции кратковременного хранения числа или команды и выполнения над ними некоторых операций. Отличается от ячейки памяти тем, что может не только хранить двоичный код, но и преобразовывать его.

Рисунки – объекты векторной природы (линии, прямые и кривые, геометрические

фигуры, стандартные и нестандартные).

С

Сайт – совокупность Web-страниц, тематически связанных между собой и расположенных на одном сервере.

Сбор информации – это совместно осуществляемые процедуры поиска и отбора информации.

Связь – такое отношение между объектами, когда изменение параметров одного объекта приводит к изменению параметров другого объекта.

Семантика – система правил и соглашений, определяющая толкование и придание смысла конструкциям языка.

Сервер – высокопроизводительный компьютер с большим объёмом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования).

Сервисное программное обеспечение – это совокупность программных продуктов, предоставляющих пользователю дополнительные услуги в работе с компьютером и расширяющих возможности операционных систем.

Сетевая информационная модель – применяется для отражения систем со сложной структурой, в которой связи между элементами имеют произвольный характер.

Сетевые приложения – усовершенствованные программы, доступные через коммуникационную систему сети.

Сеть компьютерная – совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети.

Сеть одноранговая – тип сети, где каждый компьютер может быть клиентом для других компьютеров, а также действовать в качестве сервера.

Сеть с выделенным сервером – тип сети, где требуется присутствие сервера как для обеспечения служб и ресурсов, так и для управления доступом к этим службам и ресурсам.

Сжатие – кодирование, которое обеспечивает уменьшение количества информации без потери ее содержания.

Сигнал – любой процесс, несущий информацию.

Сигнатура – это некоторая постоянная последовательность программного кода, специфичная для конкретной вредоносной программы.

Синтаксис – набор правил построения фраз языка, позволяющий определить, какие комбинации символов являются осмысленными предложениями в этом языке.

Система – совокупность взаимосвязанных элементов.

Система команд – совокупность операций, выполняемых некоторым компьютером.

Система команд исполнителя (СКИ) – это вся совокупность команд, которые исполнитель умеет выполнять (понимает).

Система счисления – это знаковая система, в которой числа записываются по определенным правилам с помощью символов некоторого алфавита, называемых

цифрами.

Системные программы – программы общего пользования, выполняемые вместе с прикладными и служащие для управления ресурсами компьютера – центральным процессором, памятью, вводом-выводом.

Системный администратор – лицо, управляющее организацией работы участников локальной компьютерной сети.

Системы автоматизированного проектирования (САПР) – комплексные программно-технические системы, предназначенные для выполнения проектных работ с применением математических методов. Широко используются в архитектуре, электронике, механике и др.

Сканер – устройство для ввода в компьютер документов - текстов, чертежей, графиков, рисунков, фотографий. Создает оцифрованное изображение документа и помещает его в память компьютера.

Слайд – электронная страница презентации.

Слоты – разъемы для подключения дополнительных устройств.

Сообщение – это информация, представленная в определенной форме и предназначенная для передачи.

Спам – массовая рассылка предварительно несогласованных электронных писем.

Спутниковые линии связи – каналы связи, которые используются для подключения к Интернету удаленных компьютерных сетей путем передачи и приема информации со спутника с использованием антенны спутниковой связи.

Стример – это устройство внешней памяти для хранения данных на магнитной ленте, по принципу напоминающее обычный кассетный магнитофон.

Схема алгоритма (блок-схема) – графическое представление алгоритма в виде последовательности блоков, соединённых стрелками.

Т

Такт – это промежуток времени между началами подачи двух последовательных импульсов специальной микросхемой – генератором тактовой частоты, синхронизирующим работу узлов компьютера.

Тактовая частота – это количество тактов в секунду.

Тачпад – сенсорная панель, реагирующая на движение пальца пользователя по поверхности. Перемещение пальца по поверхности сенсорной панели преобразуется в перемещение курсора на экране монитора. Нажатие на поверхность сенсорной панели эквивалентно нажатию на кнопку мыши.

Текстовый процессор – программное средство, предназначенное для создания, редактирования и форматирования простых и комплексных текстовых документов.

Телеконференции (группы новостей) – это глобальные или локальные форумы, в которых люди с общими интересами обмениваются информацией, обсуждают различные темы и задают друг другу вопросы.

Тест – проверка какого-либо объекта с эталоном.

Тестирование – этап решения задачи на компьютере, в процессе которого проверяется работоспособность программы, не содержащей явных ошибок.

Технические средства (hardware) – электронные и/или механические части вычислительного устройства (компьютер, ЭВМ, микроЭВМ и т.д.).

Тип данных – понятие языка программирования, определяющее структуру констант, переменных и других элементов данных, разрешенные их значения и операции, которые можно над ними выполнять.

Топология сети – способ описания конфигурации сети, схема расположения и соединения сетевых устройств.

Трафик – объем информации, передаваемой по сети за определенный период времени.

Трекбол – указательное устройство ввода информации об относительном перемещении для компьютера. Аналогично мыши по принципу действия и функциям.

У

Угроза безопасности информации – события или действия, которые могут привести к искажению, несанкционированному использованию или даже к разрушению информационных ресурсов управляемой системы, а также программных и аппаратных средств.

Упаковщики (архиваторы) – программы, позволяющие записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл.

Устройство управления (УУ) – часть процессора, выполняющая функции управления устройствами компьютера.

Утилита – сервисная программа вспомогательного назначения, предназначенная для обслуживания и повышения эффективности вычислительной системы.

Ф

Файл – это поименованная область на диске или другом машинном носителе.

Файловая система – способ хранения файлов на дисках компьютера.

Файловая структура – иерархическая структура, в виде которой операционная система отображает файлы и папки диска.

Форматирование – процесс представления внешнего вида документа или отдельных его объектов в требуемой форме.

Формула – это арифметическое или логическое выражение, содержащее константы, операторы, ссылки, функции, имена диапазонов. Ввод формулы всегда начинается со знака =.

Ц

Цветовая палитра (палитра цвета) – фиксированный набор цветов и оттенков, имеющий внешнюю или цифровую реализацию в том или ином виде.

Центральный процессор – исполнитель машинных инструкций, часть аппаратного обеспечения компьютера или программируемого логического контроллера; отвечает за выполнение операций, заданных программами.

Цикл – приём в программировании, позволяющий многократно повторять одну и ту же последовательность команд (операторов).

Циклический алгоритм – алгоритм, последовательность шагов которого выполняется многократно.

Цифровой сигнал – это сигнал, который может принимать только одно из двух установленных значений.

Ч

Чипсет – набор микросхем, установленных на материнской плате.

Чувствительный экран – позволяет осуществлять общение с компьютером путем прикосновения пальцем к определённом месту экрана монитора.

Ш

Шины – набор проводников, по которым происходит обмен сигналами между внутренними устройствами.

Шлюз – сетевое устройство или программное средство для сопряжения разнородных сетей.

Э

Эквивалентность – это логическая операция, ставящая в соответствие каждому двум простым высказываниям составное высказывание, являющееся истинным тогда и только тогда, когда оба исходных высказывания одновременно истинны или одновременно ложны.

Экспертная система – комплекс компьютерного программного обеспечения, помогающий человеку принимать обоснованные решения.

Электронная вычислительная машина (ЭВМ) — вычислительная машина, предназначенная для передачи, хранения и обработки информации.

Электронная почта – система пересылки сообщений между пользователями вычислительных систем, в которой компьютер берет на себя все функции по хранению и пересылке сообщений.

Электронные таблицы – класс программ для создания и обработки табличных данных в электронном виде.

Электронный офис – система автоматизации работы учреждения, основанная на использовании компьютерной техники.

Энтропия – мера неопределенности опыта, в котором проявляются случайные события.

Эргономичная клавиатура – клавиатура, имеющая специальную форму, рассчитанную с учетом требований эргономики.

Я

Язык – множество символов и совокупность правил, определяющих способы составления из этих символов осмысленных сообщений.

Ярлык – файл, служащий указателем на объект, программу или команду и содержащий дополнительную информацию.

Ячейка электронной таблицы – минимальный элемент для хранения данных, образующийся на пересечении столбца и строки.

ЛИТЕРАТУРА

1. Кудинов Ю. И. Основы современной информатики : учеб. пособие для вузов /Кудинов Ю. И., Пащенко Ф. Ф. - СПб. :Лань, 2011. - 256 с.
2. Макарова Н.В., Волков В.Б. Информатика: Учебник для вузов. – СПб.: Питер, 2011. – 576.
3. Могилев А. В. Информатика : учеб. пособие для вузов /Могилев А. В. , Пак Н. И., Хеннер Е. К. - М. :Академия, 2012. - 848 с. - (Высшее профессиональное образование)
4. Новожилов О. П. Информатика : учеб. пособие для бакалавров /Новожилов О. П., - М. :Юрайт, 2012. - 564 с.
5. Симонович С.В. Информатика. Базовый курс : учеб. для вузов. 3-е изд. Стандарт третьего поколения. - СПб. :Питер, 2011. - 640 с.
6. Степанов А. Н. Информатика. Базовый курс : учеб. пособие для вузов /Степанов А. Н., - СПб. :Питер, 2011. - 720 с.

Учебное издание

Петракова Наталья Васильевна

ИНФОРМАТИКА

**КУРС ЛЕКЦИЙ
по направлению подготовки бакалавров
очной формы обучения**

Редактор Павлютина И.П.

Подписано к печати 1.09.2015 г.

Формат 60x84. 1/16. Бумага печатная. Усл. печ. л. 9,99.

Тираж 100 экз. Изд. № 3187.

Издательство Брянского государственного аграрного университета

243365, Брянская обл., Выгоничский район, п. Кокино, БГАУ