

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ИЖЕВСКАЯ ГОСУДАРСТВЕННАЯ СЕЛЬСКОХОЗЯЙСТВЕННАЯ АКАДЕМИЯ»

УТВЕРЖДАЮ

Проректор по учебной и  
воспитательной работе, профессор

 С.Л. Воробьева  
« 14 » июня 2019 г.

## ИНФОРМАТИКА

Курс лекций

Учебное пособие

для студентов бакалавриата, обучающихся в аграрных вузах

Составители:

Е. С. Третьякова, Е. В. Тимошкина

Ижевск  
ФГБОУ ВО Ижевская ГСХА  
2019

УДК 004.9(078)  
ББК 32.973-018.2я73-9  
И 74

Учебное пособие подготовлено в соответствии с федеральными государственными образовательными стандартами высшего образования.

Рассмотрено и рекомендовано к изданию редакционно-издательским советом ФГБОУ ВО Ижевская ГСХА, протокол № 3 от 11.06.2019.

Рецензент:

*О. С. Федоров* – к. т. н., доцент, заведующий кафедрой эксплуатации и ремонта машин ФГБОУ ВО Ижевская ГСХА

Составители:

*Е. С. Третьякова* – доцент каф. ЭКИТ;  
*Е. В. Тимошкина* – доцент каф. ЭКИТ

**Информатика.** Курс лекций: учебное пособие для самостоятельной работы студентов бакалавриата, обучающихся в аграрных вузах / Сост. Е. С. Третьякова, Е. В. Тимошкина. – Ижевск: ФГБОУ ВО Ижевская ГСХА, 2019. – 164 с.

В учебном пособии представлены разделы информатики, охватывающие основные вопросы теории информации, функционирования аппаратного обеспечения, алгоритмизации, принципов работы различных программных продуктов, устройства вычислительных сетей и основы теории баз данных. Изложены основные приемы программирования, используемые студентами на лабораторных занятиях, а также при индивидуальной и самостоятельной работе студентов.

УДК 004.9(078)  
ББК 32.973-018.2я73-9

© ФГБОУ ВО Ижевская ГСХА, 2019  
© Третьякова Е. С., Тимошкина Е. В., сост., 2019

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
Цели и задачи дисциплины.....	5
1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ИНФОРМАЦИИ .....	6
1.1. Предмет и задачи Информатики .....	6
1.2. Основные определения .....	9
1.3. Основные свойства информации .....	11
1.4. Классификация информации .....	11
1.5. Количество информации как мера уменьшения неопределенности знаний .....	13
1.6. Единицы измерения информации .....	14
1.7. Информационные процессы .....	15
1.8. Операции с данными .....	16
1.9. Кодирование информации в компьютере.....	17
1.10. Единицы хранения данных .....	24
1.11. Арифметические основы ЭВМ.....	26
1.12. Арифметика в двоичной системе .....	31
1.13. Вопросы для самопроверки по теме 1 .....	31
2. ВВЕДЕНИЕ В АЛГЕБРУ ЛОГИКИ .....	34
2.1. Основные понятия алгебры логики .....	34
2.2. Вопросы для самопроверки по теме 2 .....	41
3. ТЕХНИЧЕСКИЕ СРЕДСТВА РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ .....	43
3.1. Этапы развития вычислительной техники .....	43
3.2. Классификация и состав ЭВМ.....	44
3.3. Поколения ЭВМ .....	47
3.4. Архитектура персонального компьютера (ПК) .....	48
3.5. Базовая аппаратная конфигурация ПК .....	49
3.6. Периферийные устройства .....	54
3.7. Вопросы для самопроверки по теме 3 .....	56
4. ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ .....	58
4.1. Программное обеспечение персонального компьютера .....	58
4.2. Операционные системы семейства Windows .....	64
4.3. Основные элементы графического интерфейса Windows .....	65
4.4. Операции с файловой системой Windows .....	67
4.5. Вопросы для самопроверки по теме 4 .....	69
5. ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ .....	70
5.1. Текстовые редакторы .....	70
5.2. Текстовый редактор WORD, назначение и возможности. Основные средства текстового редактора WORD 2010.....	71
5.3. Электронные таблицы. Назначение и классификация .....	84
5.4. Общие сведения о табличном процессоре MS Excel.....	84
5.5. Создание презентаций с помощью программы PowerPoint.....	96
5.6. Вопросы для самопроверки по теме 5 .....	102
6. ОСНОВНЫЕ ПОНЯТИЯ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ .....	104
6.1. Базы данных и системы управления ими: основные понятия и определения, реляционная модель данных .....	104
6.2. СУБД MS Access, основные этапы разработки базы данных.....	107
6.3. Вопросы для самопроверки по теме 6 .....	114
7. ОСНОВЫ ЗАЩИТЫ ИНФОРМАЦИИ.....	115
7.1. Защита информации как закономерность развития компьютерных систем .....	115
<b>Объекты и элементы защиты в компьютерных системах обработки данных .....</b>	<b>116</b>
7.2. Средства опознания и разграничения доступа к информации .....	117
7.3. Компьютерные вирусы .....	119
7.4. Антивирусные программы.....	121

7.5. Облачная антивирусная защита .....	122
7.6. Вопросы для самопроверки по теме 7 .....	123
<b>8. ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ.....</b>	<b>124</b>
8.1. Классификация вычислительных сетей.....	124
8.2. Виды сетевых ресурсов.....	124
8.3. Топология и архитектура вычислительных сетей .....	125
8.4. Программное обеспечение вычислительных сетей.....	126
8.5. Протоколы электронной почты.....	127
8.6. Коммуникационное оборудование .....	128
8.7. Сеть Интернет.....	128
8.8. Основные возможности, предоставляемые сетью Интернет.....	130
8.9. Вопросы для самопроверки по теме 8 .....	131
<b>9. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ .....</b>	<b>132</b>
9.1. Этапы решения задач на компьютере.....	132
9.2. Понятие алгоритма и его свойства. Способы задания алгоритмов.....	133
9.3. Классификация языков программирования .....	139
9.4. Основные понятия алгоритмических языков. Алфавит. Синтаксис. Семантика .....	144
9.5. Язык программирования PascalABC .....	145
9.6. Структура программы языка Pascal .....	145
9.7. Вопросы для самопроверки по теме 9 .....	158
<b>ОТВЕТЫ НА ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....</b>	<b>160</b>
<b>СПИСОК ЛИТЕРАТУРЫ.....</b>	<b>162</b>

## ВВЕДЕНИЕ

### Цели и задачи дисциплины

Целями учебной дисциплины «Информатика» являются формирование представлений о сущности информации и информационных процессов, развитие алгоритмического мышления, представляющего собой необходимую часть научного взгляда на мир, изучение современных информационных технологий, демонстрация возможности использования полученных знаний в различных сферах деятельности человека. Знание основных разделов дисциплины способствует повышению эффективности учебной деятельности студентов и их будущей профессиональной деятельности, а также положительному восприятию процесса информатизации общества.

Требования к уровню освоения содержания дисциплины.

В результате изучения дисциплины студенты должны:

***знать:***

- основные понятия и определения информатики;
- основные принципы работы современного компьютера;
- технические средства обработки информации;
- программные средства обработки информации;
- основные понятия и способы моделирования;

***уметь:***

- использовать основные программные средства и информационные системы;
- решать простые логические задачи, создавать модели решения учебных задач и реализовывать их на компьютере;
- осуществлять поиск необходимой информации в сети Интернет;
- пользоваться услугами электронной почты и поисковых систем;

***владеть:***

- современными навыками обработки информации;
- технологиями решения задач с использованием компьютера;
- приемами работы с поисковыми системами, электронной почтой; навыками работы с электронными учебными ресурсами.

# 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ИНФОРМАЦИИ

## 1.1. Предмет и задачи Информатики

### Этапы предыстории информатики



Планета Информатики (<http://inf1.info>)

Корни информатики лежат в другой науке — [кибернетике](#). Понятие «кибернетика» впервые появилось в первой половине века, когда французский физик [Андре Мари Ампер](#) решил создать единую классификацию всех наук, как существовавших в то время, так и гипотетических (которые не существовали, но, по его мнению, должны были существовать). Он предположил, что должна существовать некая наука, занимающаяся изучением искусства управления. Ампер не имел в виду управление техническими системами, поскольку сложных технических систем в те времена еще не было. Он имел в виду искусство управления людьми, то есть обществом. Эту несуществующую науку Ампер назвал *кибернетикой* от греческого слова кибернетикос (искусный в управлении). В Древней Греции этого титула удостаивались лучшие мастера управления боевыми колесницами. Впоследствии слово кибернетикос было заимствовано римлянами — так в латинском языке появилось слово *губернатор* (управляющий провинцией). Сегодня уже трудно догадаться, что слова «кибернетика» и «губернатор» имеют одно происхождение, но это так. С тех пор о кибернетике забыли более, чем на сто лет.

В 1948 году выдающийся американский математик [Норберт Винер](#), труды которого по математической логике легли в основу зарождавшегося тогда программирования, вновь возродил термин «кибернетика» и определил ее как *науку об управлении в живой природе и в технических системах*. Это определение оказалось весьма спорным. Смешивание живой природы и технических систем в одной дисциплине привело к резкому неприятию такого определения учеными многих стран. Особенно сильной критике зарождавшаяся кибернетика подверглась в Советском Союзе.

Сегодня кибернетика продолжает изучать связь между психологией и математической логикой, разрабатывает методы создания искусственного

интеллекта, но наряду с ней уже действует другая, отделившаяся от нее наука – **информатика**.

Термин информатика возник в 60-х гг. во Франции для названия области, занимающейся автоматизированной обработкой информации с помощью электронных вычислительных машин. Французский термин *informatique* (информатика) образован путем слияния слов *information* (информация) и *automatique* (автоматика) и означает "информационная автоматика или автоматизированная переработка информации".

В англоязычных странах этому термину соответствует синоним *computer science* (наука о компьютерной технике). Оттуда это название и пришло к нам в Россию, а также в некоторые другие страны Восточной Европы.

Известно *классическое определение информатики как науки изучающей структуру, общие свойства, вопросы сбора, хранения, поиска, переработки и преобразования, использования знаний научно-технической информации*.

Термин был позаимствован и с середины 1970-х гг. прочно вошел сначала в научно-технический обиход, а затем стал общеизвестным и общепринятым.

В 1978 году международный научный конгресс официально закрепил за понятием "*информатика*" области, связанные с **разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты компьютеризации — массового внедрения компьютерной техники во все области жизни людей**.

В нашей стране подобная трактовка термина "информатика" утвердилась с момента принятия решения в 1983 г. на сессии годовичного собрания Академии наук СССР об организации нового отделения информатики, вычислительной техники и автоматизации. Информатика трактовалась как "*комплексная научная и инженерная дисциплина, изучающая все аспекты разработки, проектирования, создания, оценки, функционирования основанных на ЭВМ систем переработки информации, их применения и воздействия на различные области социальной практики*".

Высшей наградой за заслуги в области информатики является премия Тьюринга.

4 декабря отмечается День российской информатики, так как в этот день в 1948 году Государственный комитет Совета министров СССР по внедрению передовой техники в народное хозяйство зарегистрировал за номером 10 475 изобретение И.С. Брука и Б.И. Рамеева — цифровую электронную вычислительную машину

Существует множество определений информатики, что связано с многогранностью ее функций, возможностей, средств и методов.

Российский академик А.А. Дородницын выделяет в информатике три неразрывно и существенно связанные части — **технические средства, программные и алгоритмические.**

**Технические средства**, или аппаратура компьютеров, в английском языке обозначаются словом **Hardware**, которое буквально переводится как "твердые изделия".

Для обозначения **программных средств**, под которыми понимается **совокупность всех программ, используемых компьютерами, и область деятельности по их созданию и применению**, используется слово **Software** (буквально — "мягкие изделия"), которое подчеркивает равнозначность самой машины и программного обеспечения, а также способность программного обеспечения модифицироваться, приспособливаться и развиваться.

Программированию задачи всегда предшествует **разработка способа ее решения в виде последовательности действий, ведущих от исходных данных к искомому результату**, иными словами, **разработка алгоритма решения задачи**. Для обозначения части информатики, связанной с разработкой алгоритмов и изучением методов и приемов их построения, применяют термин **Brainware** (англ. brain — интеллект).

**Информатика** – это наука, которая изучает структуру и общие свойства информации, а также информационные процессы в живой и неживой природе, обществе и технике.

Исходя из этого определения **информатика** – это и наука, и область практической деятельности человека.

**Предмет информатики как науки составляют:**

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

Средства взаимодействия в информатике принято называть **интерфейсом**. Поэтому средства взаимодействия аппаратного и программного обеспечения называют программно-аппаратным интерфейсом, а средства взаимодействия человека с аппаратными и программными средствами — **интерфейсом пользователя**.

В составе **основной задачи** сегодня можно выделить такие основные направления информатики для практического применения:

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

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

**Роль информатики в развитии общества чрезвычайно велика. С ней связано начало революции в области накопления, передачи и обработки информации. Эта революция, следующая за революциями в овладении веществом и энергией, затрагивает и коренным образом преобразует не только сферу материального производства, но и интеллектуальную, духовную сферы жизни.**

Прогрессивное увеличение возможностей компьютерной техники, развитие информационных сетей, создание новых информационных технологий приводят к значительным изменениям во всех сферах общества: в производстве, науке, образовании, медицине и т.д.

## **1.2. Основные определения**

**Информатика** – это дисциплина, изучающая структуру и общие свойства **информации**, закономерности и методы её создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности.

В настоящее время большинство операций с информацией совершается с помощью ЭВМ. Поэтому сведения о компьютерах и компьютерные технологии обработки информации являются важной составной частью дисциплины «информатика».

Понятие **информация** точно и однозначно не определяется, хотя используется повсеместно. Оно вводится путём объяснения, которое опирается на интуицию, здравый смысл или бытовое применение этого термина.

В Федеральном законе Российской Федерации от 27 июля 2006г. №149-ФЗ «Об информации, информационных технологиях и о защите информации» (<http://www.rg.ru/2006/07/29/informaciadok.html>) дается следующее определение этого термина: *«информация — сведения (сообщения, данные) независимо от формы их представления».*

Толковый словарь русского языка Ожегова приводит 2 определения слова «информация»:

1. Сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специальным устройством.

2. Сообщения, осведомляющие о положении дел, о состоянии чего-нибудь. (Научно-техническая и газетная информация, средства массовой информации — печать, радио, телевидение, кино).

Для количественного определения имеющейся информации самым удобным оказалось такое: *это сведения, которые уменьшают неопределенность об окружающем мире и являются объектом хранения, преобразования, передачи и использования.*

**Энтропия** – это мера неопределённости наших знаний об объекте или явлении. Энтропию иногда называют антиинформацией. Например. Если мы интересуемся сведениями о полностью засекреченном объекте или явлении, или получили зашифрованное сообщение, а ключа к расшифровке не знаем, то для нас информация о нём равна нулю, а энтропия – максимальна.

**Знания** – это осознанные и запомненные людьми свойства предметов, явлений и связей между ними, а также способов выполнения тех или иных действий для достижения нужных результатов.

**Сигнал** (сообщение) – информационный поток, который в процессе передачи информации поступает к приёмнику.

**Данные** – это зарегистрированные на материальном носителе сигналы.

Одна и та же информация может передаваться с помощью разных сообщений. Например, сведения о выпуске книги могут быть переданы с помощью телевидения, устного разговора, рекламных щитов и т. д. И, наоборот, одно и то же сообщение может нести различную информацию. Пример: сообщение на китайском языке несёт какую-то информацию только для тех, кто этот язык знает.

**Сведения, факты, данные** – это знания, выраженные в сигналах, сообщениях, известиях, уведомлениях и т.д.

**Информационные процессы** – это хранение, передача и обработка данных.

**Информационная революция** – это преобразование общественных отношений из-за кардинальных изменений в сфере обработки информации.

Другими словами, информационная революция означает скачок в развитии общества, новый уровень использования принципиально новых методов и средств переработки информации и процессов информационного взаимодействия в обществе, что создает основу для объединения интеллектуальных способностей человечества.

В настоящее время выделяется шесть основных информационных революций в истории развития человеческого общества. Они связываются со следующими событиями:

- появление человеческой речи;
- изобретение письменности;
- изобретение книгопечатания;
- изобретение радио, телефона, телевидения;
- изобретение микропроцессорных технологий и появление персональных компьютеров;

- создание компьютерных сетей и, в частности, глобальной сети Интернет.

### 1.3. Основные свойства информации

Свойства информации можно рассматривать в трех аспектах:

- **технический** – это точность, надежность, скорость передачи сигналов и т. д.;

- **семантический** – это передача смысла текста с помощью кодов. Например, при семантической отладке программы проверяются типы переменных, входящих в выражение: если переменная А текстовая, переменная В – числовая, то выражение А/В не имеет смысла, если же А и В – числовые переменные, то это выражение становится осмысленным;

- **прагматический** – это насколько эффективно информация влияет на поведение объекта.

С учетом этих факторов к основным свойствам информации относятся:

- **объективность** — не зависит от чьего-либо мнения, суждения;
- **достоверность** — отражение истинного положения дел, отсутствие скрытых ошибок;

- **полезность или релевантность** оценивается по тем задачам, которые мы можем решить с её помощью, соответствие запросам потребителя;

- **полнота** — достаточно для понимания и принятия решения;

- **актуальность или своевременность** — важность для настоящего времени;

- **понятность** — информация понятна, если она выражена на языке, доступном для получателя;

- **доступность** — возможность ее получения данным потребителем;

- **защищенность** — невозможность несанкционированного использования или изменения;

- **эргономичность** — удобство формы или объема с точки зрения данного потребителя;

- **краткость, четкость** — отсутствие в информации ненужных сведений.

### 1.4. Классификация информации

Поскольку носителями информации являются сигналы, то в качестве последних могут использоваться физические или социальные процессы различной природы. Например, процесс протекания электрического тока в цепи, процесс механического перемещения тела, количество людей на предприятии, имеющих высшее образование, продуктивность работы ученых и т. д. Сигналом, передающим информацию, служит значение одного или нескольких параметров регистрируемого процесса. В связи с этим существуют разные варианты классификации информации.

1. *По форме представления:*

- **дискретная информация:** характеризуется прерывистой, изменяющейся величиной, например, количество символов в том или ином алфавите, количество занятых байт в памяти компьютера и т. п. Сигнал, переносящий информацию, представляется последовательностью символов алфавита, принятого в данной предметной области;

- **аналоговая информация:** (непрерывная) представляется сигналом, измеряемый параметр которого может принимать любые промежуточные значения в определенных пределах. Например, температура тела человека, скорость автомобиля на определенном участке пути, воспроизведение звука на виниловой пластинке, так как звуковая дорожка на ней изменяет свою форму непрерывно, плавные переходы цветов на живописном полотне и т. п. Аналоговую информацию можно преобразовать в дискретную с некоторой потерей промежуточных значений.

Для цифровой техники наиболее удобна дискретная форма представления информации.

2. *По области возникновения выделяют информацию:*

- **механическую,** которая отражает процессы и явления неодушевленной природы;

- **биологическую,** которая отражает процессы животного и растительного мира;

- **социальную,** которая отражает процессы человеческого общества.

3. *По способу передачи и восприятия различают следующие виды информации:*

- **визуальную,** передаваемую видимыми образами и символами;

- **аудиальную,** передаваемую звуками;

- **тактильную,** передаваемую ощущениями прикосновений;

- **обонятельную,** передаваемую запахами;

- **вкусовую,** передаваемую вкусами.

4. *Информацию, создаваемую и используемую человеком, по общественному назначению можно разбить на три вида:*

- **личную,** предназначенную для конкретного человека;

- **массовую,** предназначенную для любого желающего ею пользоваться (общественно-политическая, научно-популярная и т.д.);

- **специальную,** предназначенную для использования узким кругом лиц, занимающихся решением сложных специальных задач в области науки, техники, экономики.

5. *По способам кодирования выделяют следующие типы информации:*

- **символьную,** основанную на использовании символов – букв, цифр, знаков и т. д. Она является наиболее простой, но применяется только для передачи несложных сигналов о различных событиях. Примером может

служить зеленый свет уличного светофора, который сообщает пешеходам и водителям автотранспорта о возможности начала движения;

- **текстовую**, основанную на использовании комбинаций символов. Здесь так же, как и в предыдущей форме, используются символы: буквы, цифры, математические знаки. При этом в текстовой информации принципиально важен не только состав, но и порядок следования символов. Так, слова КОТ и ТОК имеют одинаковые буквы, но содержат различную информацию. **Текстовая** информация. чрезвычайно удобна и широко используется в деятельности человека: книги, брошюры, журналы, различного рода документы, аудиозаписи кодируются в текстовой форме;

- **графическую**, основанную на использовании произвольного сочетания графических примитивов. К этой форме относятся фотографии, схемы, чертежи, рисунки, играющие большое значение в деятельности человека.

### **1.5. Количество информации как мера уменьшения неопределенности знаний**

Объём используемой человеком информации в мире постоянно растёт. В таблице 1 показана динамика роста человеческих знаний.

Таблица 1 – Увеличение человеческих знаний

<b>Общая сумма человеческих знаний удваивалась:</b>	
Каждые 50 лет	до 1800 года
Каждые 10 лет	до 1950 года
Каждые 5 лет	до 1970 года
Ежегодно	до 1990 года

Подход к информации как к мере уменьшения неопределённости наших знаний позволяет количественно измерять информацию, полученную через некоторое сообщение.

Например, после сдачи зачета Вы получаете одно из двух информационных сообщений: "зачет" или "незачет", а после сдачи экзамена одно из четырех информационных сообщений: "2", "3", "4" или "5". Информационное сообщение об оценке за зачет приводит к уменьшению неопределенности вашего знания в два раза, так как реализуется один из двух возможных вариантов. Информационное сообщение об оценке за экзамен приводит к уменьшению неопределенности вашего знания в четыре раза, так как получено одно из четырех возможных информационных сообщений.

Ясно, что чем более неопределенна первоначальная ситуация, тем больше мы получим новой информации при получении информационного сообщения о том, как она разрешилась (тем в большее количество раз уменьшится неопределенность знания).

**Клод Шеннон** предложил в 1948 году формулу для определения количества информации, которую мы получаем после получения одного из  $N$  возможных сообщений:

$$I = - (p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_i \log_2 p_i + \dots + p_N \log_2 p_N)$$

Здесь  $p_i$  – вероятность того, что будет получено именно  $i$ -е сообщение.

Если все сообщения равновероятны, то все  $p_i = 1/N$ , и из этой формулы получается формула **Хартли**:

$$I = \log_2 N$$

Для количественного выражения любой величины необходимо сначала определить единицу измерения. Так, для измерения длины в качестве единицы выбран метр, для измерения массы - килограмм и т. д. Аналогично, для определения количества информации необходимо ввести единицу измерения.

Из формулы Хартли следует: если  $I=1$ , то  $N=2$ , то есть в качестве единицы измерения информации можно взять тот объем информации, который мы получаем при принятии сигнала о том, что же произошло в ситуации с двумя возможными исходами. Такая единица названа **битом**.

### 1.6. Единицы измерения информации

**Бит** – это минимальная единица измерения количества информации.

Единица измерения информации **бит** (bit) - сокращение от английских слов **binary digit**, что означает двоичная цифра.

В компьютерной технике бит соответствует физическому состоянию носителя информации: намагничено - не намагничено, есть отверстие - нет отверстия. При этом одно состояние принято обозначать цифрой 0, а другое - цифрой 1. Выбор одного из двух возможных вариантов позволяет также различать логические величины: *истину* и *ложь*.

Более крупные единицы формируются в информатике способом, который несколько отличается от принятых в большинстве наук. Первой более крупной, чем бит, единицей измерения информации, выбран байт:

$$1 \text{ байт} = 8 \text{ бит} = 2^3 \text{ бит}$$

Для измерения более крупных объемов информации используются приставки, применяемые в традиционной международной системе единиц СИ.

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

$$1 \text{ Мегабайт} = 1024 \text{ Килобайт} = 2^{20}$$

$$1 \text{ Гигабайт} = 1024 \text{ Мегабайт} = 2^{30}$$

Для измерения больших объемов информации используются следующие единицы:

$$1 \text{ Тбайт (один терабайт)} = 2^{10} \text{ Гбайт} = 1024 \text{ Гбайт} = 2^{40} \text{ байт}$$

$$1 \text{ Пбайт (один петабайт)} = 2^{10} \text{ Тбайт} = 1024 \text{ Тбайт} = 2^{50} \text{ байт}$$

$$1 \text{ Эбайт (один эксабайт)} = 2^{10} \text{ Пбайт} = 1024 \text{ Пбайт} = 2^{60} \text{ байт}$$

$$1 \text{ Збайт (один зетабайт)} = 2^{10} \text{ Эбайт} = 1024 \text{ Эбайт} = 2^{70} \text{ байт}$$

$$1 \text{ Йбайт (один йотабайт)} = 2^{10} \text{ Збайт} = 1024 \text{ Збайт} = 2^{80} \text{ байт}$$

**Пример.** Книга содержит 100 страниц; на каждой странице - 35 строк, в каждой строке - 50 символов. Рассчитаем объем информации, содержащийся в книге.

Страница содержит  $35 \times 50 = 1750$  байт информации. Объем всей информации в книге (в разных единицах):

$$1750 \times 100 = 175000 \text{ байт.}$$

$$175000 / 1024 = 170,8984 \text{ Кбайт.}$$

$$170,8984 / 1024 = 0,166893 \text{ Мбайт.}$$

## 1.7. Информационные процессы

### 1. Хранение информации

**Хранение информации** – это процесс помещения информации в определенное хранилище с целью извлечения ее оттуда через некоторое время для дальнейшего использования

Для хранения информации человек, прежде всего, использует свою **память**. Можно считать, что мозг – это одно из самых совершенных хранилищ информации, во многом превосходящее компьютерные средства. Для запоминания и поиска информации используются нервные клетки мозга – **нейроны**, их более ста миллиардов. К сожалению, человек многое забывает, поэтому необходимо хранить информацию.

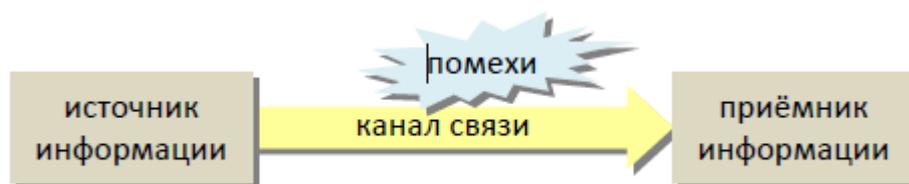
Древние: камень, папирус, береста, пергамент

Современные: бумага, магнитная лента, перфокарты и перфоленты, электронные носители: магнитные диски, лазерные диски, флэш-память

### 2. Передача (трансляция) информации

**Передача информации** – это целенаправленный процесс, в результате которого информация передается от одного объекта к другому.

Простейшая схема передачи информации



Для существования информации обязателен какой-либо материальный объект передающий или хранящий ее. Такой объект называют **носителем**.

**Носители бывают:**

а) кратковременные (ток, звуковые волны, оперативная память)

б) долговременные (бумага, ткань, жесткий диск)

Основная характеристика носителей является **разрешающая способность носителя** — максимальный объем данных передаваемых с помощью носителя.

Среда, по которой носитель перемещается в пространстве от источника к потребителю (воздух, вода, электропровод) называется **каналом связи**.

Основная характеристика канала — *пропускная способность*, т.е. максимальное количество (объем) информации, проходящее по нему в единицу времени.

В таблице приведены примеры процессов, используемых для передачи информации, и связанных с ними сигналов.

Способ передачи	Процесс	Параметры сигнала
Звук	Звуковые волны	Высота и громкость звука
Радио, телевидение	Радиоволны	Частота, амплитуда или фаза радиоволны
Изображение	Световые волны	Частота и амплитуда световых волн
Телефон, компьютерная сеть	Электрический ток	Частота и амплитуда электрических колебаний в линии связи

Однако одиночный сигнал не может содержать много информации. Поэтому для передачи информации используется ряд следующих друг за другом сигналов.

Последовательность сигналов называется **сообщением**.

Следовательно, сообщение служит переносчиком информации, а информация является содержанием сообщения

### 3. Обработка информации

**Обработка** – это любое изменение информации, причем изменяться может как содержание информации, так и ее форма.

- **Создание новой информации** — например, решение задачи с помощью вычислений или логических рассуждений.

- **Кодирование** — когда меняется *форма* (внешний вид), но не содержание информации; например, перевод текста на другой язык; один из видов кодирования – шифрование, цель которого – скрыть смысл (содержание) информации от посторонних.

- **Поиск информации** — например, в книге, в библиотечном каталоге, на схеме или в Интернете.

- **Сортировка** — расстановка элементов списка в заданном порядке, например, расстановка чисел по возрастанию или убыванию, расстановка слов или фамилий по алфавиту; одна из задач сортировки – облегчить поиск информации.

## 1.8. Операции с данными

**Данные** — это информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека. Данные – это результаты наблюдений над объектами и явлениями, которые по каким-то причинам не используются, а только хранятся. Как только данные начинают использоваться в практических целях, они превращаются в информацию. Т.о. информация – это используемые данные.

**Данные**— это совокупность сведений, зафиксированных на определенном носителе в форме, пригодной для постоянного хранения, передачи и обработки

**Итак, данными** называются зарегистрированные сигналы. Для их регистрации с целью хранения и передачи необходим некоторый язык. Этот язык должен быть понятен как отправителю информации, так и ее

получателю. Данные могут нести в себе информацию о событиях, происходящих в материальном мире. Однако данные не тождественны информации. Для получения информации нужен **метод обработки данных**.

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

Информация существует только в момент взаимодействия данных и методов. В остальное время она содержится в виде данных. Из одних и тех же данных разные получатели могут извлечь разную информацию в зависимости от адекватности методов их обработки.

Над данными можно выполнять различные операции, состав которых определяется решаемой задачей. Перечисленные ниже операции с данными не зависят от того, кто их выполняет – техническое устройство, компьютер или человек.

- **Сбор данных** – накопление данных с целью обеспечения достаточной их полноты для принятия решений.

- **Формализация данных** – приведение данных, поступающих из разных источников, к одинаковой форме, что позволяет сделать их сопоставимыми между собой.

- **Фильтрация данных** – отсеивание данных, в которых нет необходимости для принятия решений, при этом снижается уровень шума и повышается их достоверность и адекватность.

- **Сортировка данных** – упорядочение данных по заданному признаку с целью удобства использования.

- **Защита данных** – комплекс мер, направленных на предотвращение утраты, воспроизведения и модификации данных.

- **Архивация данных** – организация хранения данных в удобной и легкодоступной форме, снижающей затраты на хранение и повышающей общую надежность информационного процесса.

- **Транспортировка данных** – приём и передача данных между удаленными участниками информационного процесса.

- **Преобразование данных** – перевод данных из одной формы в другую. Часто связано с изменением носителя. Например, книги можно хранить в бумажной форме, а можно в электронном виде.

## 1.9. Кодирование информации в компьютере

**Кодирование** — это отображение дискретного сообщения в виде определенных сочетаний символов. Совокупность правил, по которым выполняется кодирование, называется *кодом* (от французского слова code-кодекс, свод законов). Другими словами, *код*— это правило отображения информации.

Благодаря кодированию компьютер может обрабатывать различного рода информацию: числовую, текстовую, графическую, звуковую, видео. Все эти виды информации после кодировки приводятся к одному виду –

последовательности электрических импульсов, в которой наличие импульса обозначается единицей, а его отсутствие – нулем.

Такой метод представления информации называется **двоичным кодированием**. Таким образом, **двоичный код является универсальным средством кодирования информации**.

#### **Кодирование чисел**

Поскольку компьютер оперирует с двоичными числами, любая вводимая в него информация должна быть представлена в виде двоичного кода. Например, когда пользователь вводит с клавиатуры десятичные числа, они сразу преобразуются в двоичные числа (это процесс кодирования). Над этими числами компьютер выполняет необходимые арифметические операции. Полученный результат компьютер может вывести на экран монитора или на принтер. Чтобы пользователь смог понять выводимую информацию, числа должны быть снова представлены в десятичной системе счисления (процесс декодирования).

#### **Кодирование текстовой информации**

Если каждому символу алфавита сопоставить определенное целое число (например, порядковый номер), то с помощью двоичного кода можно кодировать и текстовую информацию. Для хранения двоичного кода одного символа выделен **1 байт = 8 бит**.

Первоначально для кодирования одного символа использовался 1 байт. Т.к. каждый бит принимает значение 0 или 1, количество их возможных сочетаний в байте равно  $2^8=256$ . Эти состояния перенумерованы, и каждому сопоставляется какойлибо буквенный символ, графический элемент или команда, необходимая при оформлении текстовой информации. Такое соответствие называется кодовой таблицей.

В настоящее время существуют и применяются разные варианты 8-битных кодовых таблиц. Наиболее популярные из них:

**ASCII** – American Standart Code for Information Interchange – американский стандартный код для обмена информацией;

**КОИ8-Р** – Код Обмена Информацией 8-битный с кириллицей;

**CP1251** – (Code Page) – кодировка с кириллицей в Microsoft Windows;

**CP866** – кодировка MSDOS;

**ISO 8859-5** – International Standards Organization – Международная организация по стандартизации. Ещё один стандарт для кодов для кириллицы.

Множество кодовых таблиц вызвано тем, что с учетом разнообразия естественных языков и фирм, выпускающих программное обеспечение, 256 состояний одного байта недостаточно для того, чтобы закодировать все встречающиеся символы и способы форматирования текста. **При разработке всех кодовых таблиц использовано следующее соглашение: первая половина таблицы – это коды с 0 по 127 – интернациональна, т. е. одинакова во всех вариантах кодировок. Первые 33 состояния (0–32) – это коды операций с текстом (перевод на новую строку, пробел, удаление**

последнего символа и т. п.). Затем состояния с 33 по 127 – это коды знаков препинания, арифметических действий, цифр, строчных и прописных букв латинского алфавита. Вторая половина кодовых таблиц отводится под знаки **национальных** и **специальных** алфавитов и ввода в текст графических элементов для оформления таблиц.

В конце 90-х годов появился новый международный стандарт **Unicode**, который отводит под символ 2 байта. Каждый блок из 2-х байт может находиться в  $2^{16} = 65536$  состояниях. Этого достаточно, чтобы в одной таблице собрать символы большинства алфавитов мира. Правда, длина текста удваивается, и скорость его обработки замедляется. Но, в связи с существенным увеличением памяти и быстродействия современных компьютеров, этот факт несущественен.

### **Кодирование графической информации**

Графическую информацию можно представлять в двух формах: **аналоговой** и **дискретной**. **Живописное полотно**, созданное художником, – это **пример аналогового представления**, а изображение, **напечатанное при помощи принтера**, состоящее из отдельных (элементов) точек разного цвета, – это **дискретное представление**.

Путем разбиения графического изображения (дискретизации) происходит преобразование графической информации из аналоговой формы в дискретную. При этом производится кодирование – присвоение каждому элементу графического изображения конкретного значения в форме кода. Создание и хранение графических объектов возможно в нескольких видах – в виде **векторного**, **фрактального** или **растрового** изображения. Отдельным предметом **считается 3D (трехмерная) графика**, в которой сочетаются векторный и растровый способы формирования изображений.

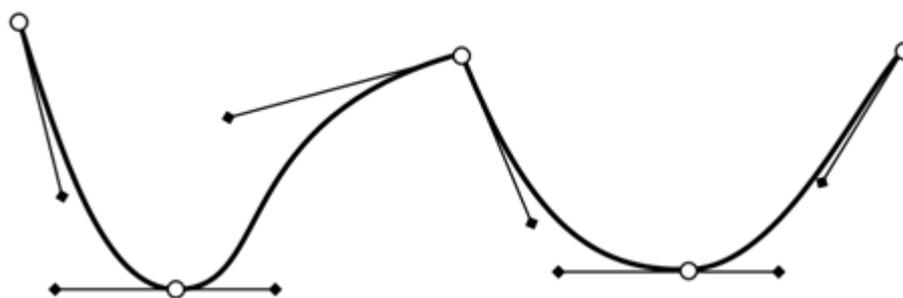
### **Векторная графика**

Для чертежей, схем, карт применяется другой способ кодирования, который позволяет не терять качество при изменении размеров изображения. Рисунок хранится как набор простейших геометрических фигур (*графических примитивов*): линий, многоугольников, сглаженных кривых, окружностей, эллипсов. Такой рисунок называется *векторным*.

**Векторный рисунок** – это рисунок, который закодирован в виде набора простейших геометрических фигур, параметры которых (размеры, координаты вершин, углы наклона, цвет контура и заливки) хранятся в виде чисел.

Векторный рисунок можно «разобрать» на части, растащив мышкой его элементы, а потом снова собрать полное изображение.

При векторном кодировании для отрезка хранятся координаты его концов, для прямоугольников и ломаных – координаты вершин. Окружность и эллипс можно задать координатами прямоугольника, в который вписана фигура. Сложнее обстояло дело со сглаженными кривыми. На рисунке изображена линия с опорными точками.



У каждой из этих точек есть «рукоятки» (*управляющие линии*), перемещая концы этих рукояток можно регулировать наклон касательной и кривизну всех участков кривой. Если обе рукоятки находятся на одной прямой, получается сглаженный узел, если нет – то угловой узел. Таким образом, форма этой кривой полностью задается координатами опорных точек и координатами рукояток. Кривые, заданные таким образом, называют *кривыми Безье* в честь их изобретателя французского инженера Пьера Безье.

Векторный способ кодирования рисунки обладает значительными **преимуществами** в сравнении с растровым тогда, когда изображение может быть полностью разложено на простейшие геометрические фигуры (например, чертеж, схема, карта, диаграмма). В этом случае при кодировании **нет потери информации**.

Объем файлов напрямую зависит от сложности рисунка – чем меньше элементов, тем меньше места занимает файл. Как правило, векторные рисунки значительно меньше по объему, чем растровые.

При изменении размера векторного рисунка не происходит никакого искажения формы элементов, при увеличении наклонных линий не появляются «ступеньки», как при растровом кодировании:

Среди форматов векторных рисунков отметим следующие:

- **WMF** (англ. *Windows Metafile* – метафайл *Windows*, файлы с расширением **.wmf** и **.emf**) – стандартный формат векторных рисунков в операционной системе *Windows*;
- **CDR** (файлы с расширением **.cdr**) – формат векторных рисунков программы *CorelDRAW*;
- **AI** (файлы с расширением **.ai**) – формат векторных рисунков программы *Adobe Illustrator*;
- **SVG** (англ. *Scalable Vector Graphics* – масштабируемые векторные изображения, файлы с расширением **.svg**) – векторная графика для веб-страниц.

Основным **недостатком** векторной графики является **невозможность изображения фотографического качества**. В векторном формате изображение всегда будет выглядеть, как рисунок.

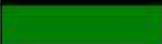
### **Растровая графика**

Дискретное представление графики получается за счет того, что экран монитора разбивается на строки и колонки. Совокупность получившихся клеточек (точек) называется растром, каждая точка – **пикселем**. Количество

строк и колонок в растре – это разрешение (разрешающая способность) экрана. Типовые разрешения: 1024\*768, 800\*600 пикселей. Первым указывается количество колонок, вторым – количество строк в растре.

Для монохромных изображений общепринятым считается кодирование цвета одного пикселя в 1байте. Это позволяет передать 254 оттенка серого плюс черный и белый цвета (всего 256 вариантов). Цветные изображения могут кодироваться разными способами в зависимости от того, для какой цели создаётся рисунок.

Метод (система) RGB (True color) – от слов Red, Green, Blue удобен для изображений, рассматриваемых на экране, выводимых на устройство записи на киноплёнку. Оттенки цвета создаются смешением лучей трёх базовых цветов разной интенсивности. Под значение интенсивности каждого луча отводится 1 байт, т. е. различают 256 уровней интенсивности. Для совокупности трёх лучей получается  $256^3=16\ 777\ 216\ 17$  млн. разных вариантов, каждый из которых создает свой оттенок цвета.

Красный	Зеленый	Синий	Название	Цвет
0	0	0	Черный	
0	1	0	Зеленый	
0	0	1	Синий	
1	0	0	Красный	
0	1	1	Бирюзовый	
1	1	0	Желтый	
1	0	1	Малиновый	
1	1	1	Белый	

Метод (система) CMYK – от слов голубой (Cyan), пурпурный (Magenta), жёлтый (Yellow), чёрный (black): удобен для изображений, которые предполагается печатать на бумаге. Он учитывает особенности полиграфии, в которой цвет получается смешением четырёх красок. Для кодирования одного пикселя требуется 4 байта и можно передать 256<sup>4</sup> 4 млрд. оттенков.

Для WEB-документов учитывать такое обилие оттенков неудобно, так как это приводит к файлам очень больших размеров, и их неудобно пересылать по сети. Поэтому в них используются так называемые индексированные цвета: из всего обилия возможных комбинаций выбрано 256 базовых оттенков. Это позволило для запоминания цвета каждого пикселя использовать только 1 байт. Каждому состоянию байта сопоставляется определенная комбинация интенсивностей базовых цветов.

Существует много разных форматов растровых рисунков. Чаще всего встречаются следующие:

- **BMP** (англ. *bitmap* – битовая карта, файлы с расширением **.bmp**) – стандартный формат в операционной системе *Windows*; поддерживает кодирование с палитрой и в режиме истинного цвета;
- **JPEG** (англ. *Joint Photographic Experts Group* – объединенная группа фотографов-экспертов, файлы с расширением **.jpg** или **.jpeg**) – формат, разработанный специально для кодирования фотографий;
- **GIF** (англ. *Graphics Interchange Format* – формат для обмена изображениями, файлы с расширением **.gif**) – формат, поддерживающий только кодирование с палитрой (от 2 до 256 цветов);
- **PNG** (англ. *Portable Network Graphics* – переносимые сетевые изображения, файлы с расширением **.png**) – формат, поддерживающий как режим истинного цвета, так и кодирование с палитрой.

**Фрактал** – это объект, отдельные элементы которого наследуют свойства родительских структур. Поскольку более детальное описание элементов меньшего масштаба происходит по простому алгоритму, описать такой объект можно всего лишь несколькими математическими уравнениями. Фракталы позволяют описывать изображения, для детального представления которых требуется относительно мало памяти.

**Трёхмерная графика (3D)** оперирует с объектами в трёхмерном пространстве. Трёхмерная компьютерная графика широко используется в кино, компьютерных играх, где все объекты представляются как набор поверхностей или частиц. Всеми визуальными преобразованиями в 3D-графике управляют с помощью **операторов, имеющих матричное представление.**

### **Кодирование аудио информации**

Музыка, как и любой звук, является не чем иным, как звуковыми колебаниями, зарегистрировав которые, её можно достаточно точно воспроизвести. Для представления звукового сигнала в памяти компьютера, необходимо поступившие акустические колебания представить в цифровом виде, то есть преобразовать в последовательность нулей и единиц. С помощью микрофона звук преобразуется в электрические колебания, после чего можно измерить амплитуду колебаний через равные промежутки времени (несколько десятков тысяч раз в секунду), используя специальное устройство – аналого-цифровой преобразователь (АЦП). Для воспроизведения звука цифровой сигнал необходимо превратить в аналоговый с помощью цифро-аналогового преобразователя. Оба эти устройства встроены в *звуковую карту* компьютера. Программное обеспечение компьютера в настоящее время позволяет непрерывный звуковой сигнал преобразовывать в последовательность электрических импульсов, которые можно представить в двоичной форме. Указанная последовательность превращений представлена на схеме:

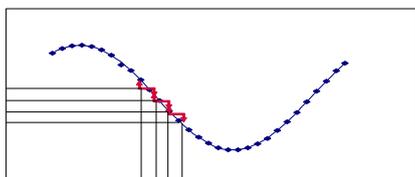
**Звуковая волна → Микрофон → Переменный электрический ток →  
→ Аудиоадаптер → Двоичный код → Память компьютера**

Процесс воспроизведения звуковой информации, сохраненной в памяти компьютера

Память компьютера → Двоичный код → Аудиоадаптер →  
→ Переменный электрический ток → Динамик → Звуковая волна

Каждое измерение звука записывается в двоичном коде. Этот процесс называется **дискретизацией (семплированием)**, выполняемым с помощью АЦП.

**Семпл** (sample англ. образец) - это промежуток времени между двумя измерениями амплитуды аналогового сигнала. Кроме промежутка времени семплом называют также любую последовательность цифровых данных, которые получили путем аналого-цифрового преобразования. Важным параметром **семплирования** является частота - количество измерений амплитуды аналогового сигнала в секунду. Диапазон частоты дискретизации звука от 8000 до 48000 измерений за одну секунду.



Графическое представление процесса дискретизации

На качество воспроизведения влияют **частота дискретизации и разрешение** (размер ячейки, отведённой под запись значения амплитуды). Например, при записи музыки на компакт-диски используются 16-разрядные значения и частота дискретизации 44032 Гц.

На слух человек воспринимает звуковые волны, имеющие частоту в пределах от 16 Гц до 20 кГц (1 Гц - 1 колебание в секунду).

В формате компакт-дисков Audio DVD за одну секунду сигнал измеряется 96 000 раз, т.е. применяют частоту семплирования 96 кГц. Для экономии места на жестком диске в мультимедийных приложениях довольно часто применяют меньшие частоты: 11,22, 32 кГц. Это приводит к уменьшению слышимого диапазона частот, а, значит, происходит искажение того, что слышно.

*Звуковой файл* - файл, хранящий звуковую информацию в числовой двоичной форме.

Звуковые файлы имеют несколько форматов. Наиболее популярные из них:

- **MIDI**
- **WAV**
- **MP3.**

Формат **MIDI** (MusicalInstrumentDigitalInterface) изначально был предназначен для управления музыкальными инструментами. В настоящее

время используется в области электронных музыкальных инструментов и компьютерных модулей синтеза

### 1.10. Единицы хранения данных

Идея представить любую информацию в виде чисел и закодировать их байтами очень рациональна. Компьютеру удобно работать, когда тексты, звуки, рисунки и видеофильмы представлены в виде байтов со значениями от 0 до 255. Непонятно только, как он отличает, где и что записано. Для этого каждая последовательность байтов, содержащая информацию определенного типа, должна быть зарегистрирована. После регистрации эта последовательность получает уникальное имя и называется *файлом*. Мы уже говорили, что наименьшей единицей *представления* информации является бит. Наименьшей единицей *обработки* или *передачи* информации является байт. Теперь мы узнали наименьшую единицу *хранения* информации – это файл. Ни байт, ни бит нельзя сохранить в качестве информации, поскольку непонятно, что они обозначают (то ли буквы, то ли ноты, то ли еще что-то). Файл можно сохранить, потому что он регистрируется, даже если в нем только байт.

**Файл** — это наименьшая единица хранения информации, содержащая последовательность байтов и имеющая уникальное имя.

Имя файла состоит из собственно имени, содержащего от 1 до 256 символов, и расширения (необязательное), состоящего из 1, 2, 3 или большего числа символов. Расширение имени файла, как правило, указывает на то, к какому типу относится его содержимое, например:

**.txt, docx** – файл содержит текст

**.pas** – в файле содержится программа на языке ПАСКАЛЬ

**.xlsx** – в файле содержится таблица, созданная программой Excel

**.exe, .com** – файлы, представляющие собой готовые к исполнению программы.

По имени файла компьютер определяет, где файл находится, какая информация в нем содержится, в каком формате она записана и какими программами ее можно обработать. Обычно в отдельном файле хранят данные, относящиеся к одному типу. В этом случае тип данных определяет *тип файла*. Чтобы данные можно было не только записать на жесткий диск, а потом еще и прочитать, надо точно знать, что и куда было записано. У всех данных должен быть *адрес*. Каждый файл на диске имеет свой адрес. Если нам нужна какая-то информация, компьютер находит на диске нужный файл, а потом байт за байтом считывает из него данные в оперативную память, пока не дойдет до конца файла. Чтобы у каждого файла на диске был свой адрес, диск разбивают на *дорожки*, а дорожки, в свою очередь, разбивают на *секторы*. Размер каждого сектора стандартен и равен 512 байтам.

*Разбиение диска на дорожки и секторы называется форматированием диска*. Его выполняют служебные программы. Самая первая дорожка магнитного диска (нулевая) считается служебной – там

хранится служебная информация, например, *таблица размещения файлов* (FAT – таблица), в которой компьютер запоминает адреса записанных файлов. Когда нам нужен какой-то файл, компьютер по его имени находит в этой таблице номер дорожки и номер сектора, после чего магнитная головка переводится в нужное положение, файл считывается и направляется в оперативную память для обработки.

Обычный жесткий диск – это устройство *физическое*. Его можно установить или удалить. *Логический жесткий диск* нельзя потрогать руками – физически он не существует. Это просто один из разделов физического диска. Каждый диск, присутствующий на компьютере имеет уникальное имя. Неважно, что это за диск: физический или логический. *Имя диска состоит из одной буквы английского алфавита и двоеточия*. Общепринятые обозначения: **A:** и **B:** — дисковод для гибких дисков, для жестких дисков имена начинаются с имени **C:** затем **D:** и так далее.

Имена файлов регистрируются на дисках в *каталогах (папках)* — это специальное поименованное место на диске, в котором хранятся файлы и другие каталоги (они называются *подкаталогами*) или может быть пустой. Каталог, с которым в данный момент работает пользователь, называется *текущим*. На каждом диске имеется один главный или *корневой* каталог. В нем регистрируются файлы и каталоги 1-го уровня. В каталогах 1-го уровня регистрируются файлы и каталоги 2-го уровня и т.д. Получается *иерархическая (древовидная) структура* каталогов на диске. В системе Windows каталогам соответствуют *папки*. В операционной системе есть специальные папки, которым не соответствуют каталоги диска. Папка – более широкое понятие. У каждой папки есть имя. **Файловую структуру** компьютера можно увидеть с помощью специальной программы, которая называется **Проводник**.

Путь к файлу. Чтобы обеспечить доступ к файлу, нужно указать путь к данному объекту по древовидной структуре каталогов. Фактически путь записывается как последовательность всех каталогов (папок) на пути от корневого каталога к нужному объекту, где имена каталогов разделяются наклонной чертой. Например: *C:\artemka\Данные\урок\_1.doc*

Полное имя документа (или его адрес) нужно для того, чтобы точно указать, на каком диске, в какой папке его найти, аналогично тому, как у нас указывается почтовый адрес человека: область, город, улица, дом, квартира и только потом фамилия.

Маски (шаблоны) - имена документов, в которых используются символы "\*" и "?". *Шаблон (маска)* — это такая фраза, с которой в дальнейшей работе будут сравниваться все имена файлов.

Символ "?" заменяет один любой символ (в том числе может обозначать и отсутствие символа). Например, маска text?.doc означает все документы с именами text1.doc, text2.doc, texta.doc, text.doc и т.д.

Символ "\*" заменяет любое число любых символов.

Например, маска *информ\*.doc* означает все документы с именами

- информатика.doc,
- информация для меня.doc,
- информация1.doc,
- информация2.doc и т.д.

Маска \*.exe означает все файлы с любыми именами и расширением .exe.

Маска text.\* означает все документы с именем text и любыми расширениями.

Маска \*.\* означает все файлы (т.е. с любыми именами и расширениями).

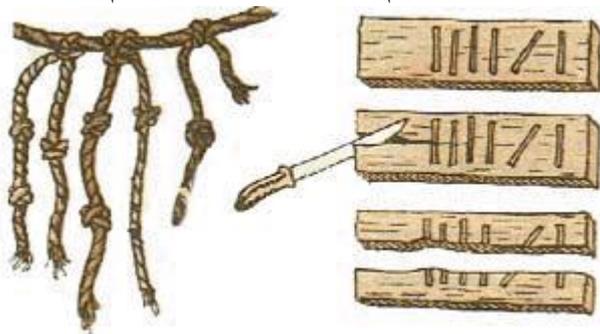
Маски удобно использовать, если Вы не уверены в правильности написания имени нужного Вам документа или хотите найти все документы какого-то одного типа и выполнить над ними какую-то операцию.

Например, по команде удалить из текущего каталога файлы \*.\* из него будут удалены все файлы.

### 1.11. Арифметические основы ЭВМ

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

Все системы счисления делятся на две большие группы: позиционные и непозиционные. Непозиционные системы счисления

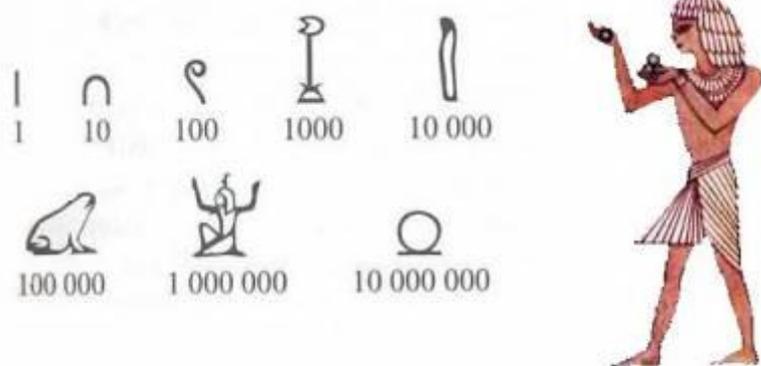


Люди научились считать очень давно. В последствии появилась потребность в записи чисел. Количество предметов изображалось нанесением черточек, засечек на какой-нибудь твердой поверхности. Чтобы два человека могли точно сохранить некоторую числовую информацию, они брали деревянную бирку, делали на ней нужное число зарубок, а потом раскалывали бирку пополам. Каждый уносил свою половинку и хранил ее. Этот прием позволял избегать спорных ситуаций. Археологами найдены такие записи при раскопках. Они относятся к 10-11 тысячелетию до н.э. Ученые назвали такую систему записи чисел единичной (унарной), так как любое число в ней образуется путем повторения одного знака, символизирующего единицу.

Позднее эти значки стали объединять в группы по 3, 5 и 10 палочек. Поэтому возникали более удобные системы счисления.

Примерно в третьем тысячелетии до нашей эры египтяне придумали свою числовую систему, в которой для обозначения ключевых чисел

использовались специальные значки – иероглифы. Каждый такой иероглиф мог повторяться не более 9 раз. Такая система счисления называется древнеегипетская десятичная непозиционная система счисления.



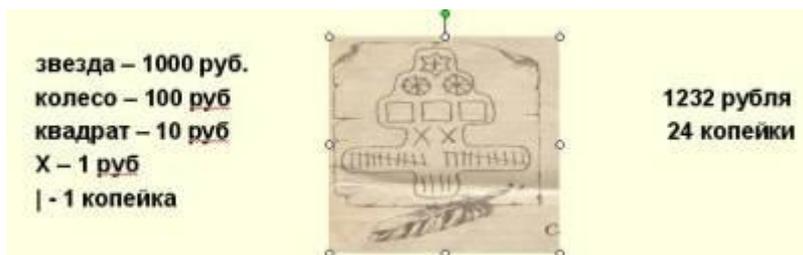
Примером непозиционной системы счисления, которая сохранилась до наших дней, может служить система счисления, применявшаяся более двух с половиной тысяч лет назад в Древнем Риме. Она называется **римская система счисления**.

В основе лежат знаки I(1), V(5), X(10), L(50), C(100), D(500), M(1000).

Римскими цифрами пользовались очень долго, сегодня они используются в основном для наименования знаменательных дат, томов, разделов и глав в книгах.

В старину на Руси широко применялись системы счисления, напоминающие римскую. Они назывались **ясачные**. С их помощью сборщики податей заполняли квитанции об уплате подати (ясака) и делали записи в податной тетради.

«Русская книга податей»



В непозиционных системах вес цифры (т.е. тот вклад, который она вносит в значение числа) не зависит от ее позиции в записи числа. В римской системе в качестве цифр используются латинские буквы:

I V X L C D M

1 5 10 50 100 500 1000

Например, VI = 5 + 1 = 6, а IX = 10 - 1 = 9. Так, в римской системе счисления в числе XXXII (тридцать два) вес цифры X в любой позиции равен просто десяти. В позиционных системах счисления вес каждой цифры изменяется в зависимости от ее положения (позиции) в последовательности цифр,

изображающих число. Например, в числе 757,7 первая семерка означает 7 сотен, вторая – 7 единиц, а третья – 7 десятых долей единицы.

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

Любая позиционная система счисления характеризуется своим основанием. *Основание позиционной системы счисления — это количество различных знаков или символов, используемых для изображения цифр в данной системе*

К позиционным системам счисления относятся:

**десятичная**  $q = 10, k = 0, 1, 2, \dots, 9$

**двоичная**  $q = 2, k = 0, 1;$

**восьмеричная**  $q = 8, k = 0, 1, 2, \dots, 7$

**шестнадцатеричная**  $q = 16, k = 0, 1, \dots, 9, A, B, C, D, E, F$

где  $q$  – основание системы счисления

$k$  – числовые знаки, используемые в системе

**Позиционные системы счисления** – это системы, в которых значение цифры зависит от её положения (позиции) в числе.

$$55,5 = 5 \cdot 10^1 + 5 \cdot 10^0 + 5 \cdot 10^{-1}$$

10 – основание десятичной системы счисления.

В степенной ряд можно разложить число, записанное в любой системе счисления. Запишем формулу разложения в общем виде.

$$K_n K_{n-1} \dots K_1 K_0, K_{-1} K_{-2} \dots K_{-m} = K_n * q^n + K_{n-1} * q^{n-1} + \dots + K_1 * q + K_0 * q^0 + K_{-1} * q^{-1} + \dots + K_{-m} * q^{-m}$$

$K$  – цифры числа

$q$  – основные системы счисления

Следовательно, для перевода чисел в десятичную систему существует правило:

*Правило1. Для того, чтобы перевести число из двоичной, восьмеричной или другой с/с в десятичную необходимо данное число разложить в ряд по степеням основания системы счисления.*

Рассмотрим обратный перевод.

*Правило2. Чтобы перевести целое десятичное число в двоичную или восьмеричную системы счисления, надо данное число разделить на основание новой системы, полученное частное снова разделить на это основание и т.д. до тех пор пока частное не будет меньше делителя. Последнее частное дает старшую цифру числа в новой системе, остальными цифрами будут остатки от деления, взятые в порядке, обратном их получению.*

$$86 \underset{\cdot}{\text{L}} 2 \rightarrow 1010110_2$$

$$\begin{array}{r} 86 \\ \underline{86} \\ 0 \end{array} \quad \begin{array}{r} 43 \text{L} 2 \\ \underline{42} \\ 1 \end{array} \quad \begin{array}{r} 21 \text{L} 2 \\ \underline{20} \\ 1 \end{array}$$

$$0 \quad \underline{42} \quad 21 \text{L} 2$$

$$189 \underset{\cdot}{\text{L}} 8 \rightarrow 275_8$$

$$\begin{array}{r} 189 \\ \underline{16} \\ 23 \end{array} \text{L} 8$$

$$\begin{array}{r} -29 \\ \underline{16} \\ 2 \end{array}$$

$$\begin{array}{r}
 1 \underline{20} \ 10 \underline{2} \\
 \text{BD}_{16} \\
 1 \ \underline{10} \ 5 \underline{2} \\
 \quad 0 \ \underline{4} \ 2 \underline{2} \\
 \quad \quad 1 \ \underline{2} \ 1 \\
 \quad \quad \quad 0
 \end{array}
 \qquad
 \begin{array}{r}
 \underline{24} \ 7 \\
 5 \\
 249_{10} \rightarrow 11111001_2 \\
 371_8 \ F \ 9_{16} \underline{16}
 \end{array}
 \qquad
 \begin{array}{r}
 189 \underline{16} \\
 \underline{16} \ 11 \ B \\
 29 \\
 13 \ D
 \end{array}$$

Правило 3. Чтобы перевести дробную часть числа из десятичной с/с в двоичную или другую с/с. Надо данную дробную часть последовательно умножать на основание новой системы счисления умножаются только дробные части числа. Дробь в новой системе запишется в виде целых частей получаемых произведений, начиная с первого сомножителя.

$$0,3125_{10} \rightarrow 0,0101_2$$

$$0,431_{10} \rightarrow 0,01101_2$$

$$\begin{array}{r}
 0, * 3125 \\
 \hline
 2 \qquad \qquad 8
 \end{array}
 \qquad
 0,431_{10} \rightarrow 0,33453_8$$

$$0, * 431 \qquad 0, * 431$$

$$\begin{array}{r}
 0 * 6250 \\
 \hline
 2 \qquad \qquad 8
 \end{array}
 \qquad
 0,431_{10} \rightarrow 0,6E56_{16}$$

$$0 * 862 \qquad 3 * 448$$

$$\begin{array}{r}
 1 * 2500 \\
 \hline
 2 \qquad \qquad 8
 \end{array}$$

$$1 * 724 \qquad 3 * 584$$

$$\begin{array}{r}
 0 * 5000 \\
 \hline
 2 \qquad \qquad 8
 \end{array}$$

$$1 * 448 \qquad 4 * 672$$

$$\begin{array}{r}
 1 \ 0000 \\
 \hline
 2 \qquad \qquad 8
 \end{array}$$

$$8 * 896 \qquad 5 * 376$$

$$1 \ 792 \qquad 3 \ 008$$

При переводе неправильных дробей отдельно переводят целую и дробную части.

$$153,34_{10} \rightarrow 10011001,01010$$

Для более быстрого перевода чисел в 2, 8 и 16 с/с используют таблицу соответствия.

**Таблица соответствия**

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	000	0	0
1	001	1	1
2	010	2	2
3	011	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Чтобы перевести число из 16 (8) системы. в 2 надо каждой шестнадцатеричной (восьмеричной) цифре поставить в соответствие четыре (три) разряда по таблице.

Чтобы перевести число из 2 в 16 надо число разбить по 4 (3) разряда начиная от запятой. И каждой тетраде (триаде) поставить в соответствие шестнадцатеричную (восьмеричную) цифру по таблице.

Между различными системами счисления существует определенная математическая связь, что позволяет осуществлять действия с членами в любой системе счисления и получать верный результат.

0, \* 431      \* 376  
      16   16  
+2 586    2 256  
  4  31  3  76  
6 \* 896    6 \* 016  
      16   16  
5 376            96  
  8  96   16  
14 \* 336    0 256  
      16  
2 016  
  3   36  
5 376

Арифметические действия в различных системах счисления. Двоичная система счисления используется для представления числовой информации в памяти ЭВМ, а 8-я и 16-я для сокращенной записи двоичных кодов.

## 1.12. Арифметика в двоичной системе

Арифметика в двоичной системе проста и, следовательно, легко реализуется в вычислительных машинах.

Таблица арифметических действий в двоичной системе счисления.

Сложение	Вычитание	Умножение
$0 + 0 = 0$	$0 - 0 = 0$	$0 * 0 = 0$
$1 + 0 = 1$	$1 - 0 = 1$	$1 * 0 = 0$
$0 + 1 = 1$	$1 - 1 = 0$	$0 * 1 = 0$
$1 + 1 = 10$	$10 - 1 = 1$	$1 * 1 = 1$

При сложении в двоичной системе 2-х единиц в данном разряде будет 0 и появится перенос единицы в старший разряд.

При вычитании из нуля единицы производится заём единицы из старшего разряда. Если в старшем разряде 0, то заём осуществляется в ближайшем старшем разряде, где есть 1. Единица, занятая в этом разряде, даёт 2 единицы в младшем разряде, где вычисляется действие, а также по единице, во всех разрядах между данным разрядом и младшим.

Арифметические действия в двоичной системе выполняются так же, как и в десятичной. Но если в 10-й перенос и заём осуществляется по 10 единиц, то в 2-й по две.

**Примеры:**

### 1. Сложение.

$$\begin{array}{r} + 1001100101, 011 \\ 10110110, 010 \quad 101110, 10 \\ \hline 1100011011, 101 \end{array} \qquad \begin{array}{r} + 10111011, 011 \\ 11101001, 111 \end{array}$$

### 2. Вычитание.

$$\begin{array}{r} - 11011101101, 011 \\ 10101110, 001 \quad 1011011, 101 \\ \hline 11000111111, 010 \end{array} \qquad \begin{array}{r} - 110110110, 110 \\ 101011011, 011 \end{array}$$

## 1.13. Вопросы для самопроверки по теме 1

**Задание № 1.** Определите, сколько бит содержит сообщение: **На улице идет дождь.**

**Задание № 2.** Определите, сколько байт может содержать сообщение: **Вылет самолёта задерживается.**

**Задание № 3.** Выберите верное утверждение:

1. в качестве материального носителя информации могут выступать знания, сведения или сообщения;

2. в качестве носителя информации могут выступать только световые и звуковые волны;
3. в качестве носителя информации могут выступать материальные предметы;
4. информационные процессы являются материальным носителем информации.

*Задание №4.* Выберите верное утверждение: свойство информации, заключающееся в достаточности данных для принятия решения, есть:

1. объективность;
2. полнота;
3. содержательность;
4. достоверность.

*Задание №5.* Выберите верное утверждение: информация достоверна, если она:

1. полезна;
2. отражает истинное положение дел;
3. достаточна для принятия решений;
4. используется в современной системе обработки информации.

*Задание №6.* Выберите верное утверждение: **сообщением в теории кодирования является:**

1. электрический импульс, распространяемый в канале связи телефонной линии;
2. воспринятая, осознанная и ставшая личностно значимой информация;
3. набор данных, объединённых смысловым содержанием и пригодных для обработки и передачи по каналам связи;
4. процесс переноса или копирования данных по некоторым признакам с одного места на другое с целью сортировки, формирования результирующих документов.

*Задание №7.* Выберите верное утверждение: в вычислительной технике в качестве основной используется \_\_\_\_\_ система счисления:

1. шестнадцатеричная;
2. десятичная;
3. восьмеричная;
4. двоичная.

*Задание №8.* Выберите верное утверждение: сканирование книги является операцией \_\_\_\_\_ данных:

1. верификации;
2. преобразования;
3. архивирования;
4. транспортировки.

*Задание №9.* Выберите верное утверждение: Стандартным кодом для обмена информацией является код ...

1. ASCII
2. ACCESS

3. ANSI

4. BIOS

*Задание № 10.* Представьте десятичное число 1023 в двоичной системе.

*Задание № 11.* Расположите в возрастающей последовательности следующие единицы измерения информации: 1Кб, 1Мб, 1Тб, 1Гб.

*Задание № 12.* Определить информационный объём фразы «Я помню чудное мгновенье» при алфавитном подходе к определению количества информации, если считать, что информационная емкость буквы русского алфавита равна 5 битам.

*Задание № 13.* Сигналы, зарегистрированные на материальном носителе, это:

1. предикаты;
2. данные;
3. байты;
4. умозаключения.

*Задание № 14.* Укажите, к какому типу относится информация, представленная в виде слов

## 2. ВВЕДЕНИЕ В АЛГЕБРУ ЛОГИКИ

### 2.1. Основные понятия алгебры логики

Трудно не согласиться с высказыванием автора детектива про Шерлока Хомса.

"...по одной капле воды...человек, умеющий мыслить логически, может сделать вывод о существовании Атлантического океана или Ниагарского водопада, даже если он не видел ни того, ни другого и никогда о них не слышал..."

По ногтям человека, по его рукам, обуви, сгибу на коленях, по выражению лица и обшлагам рубашки – по таким мелочам нетрудно угадать его профессию.

И можно не сомневаться, что все это вместе взятое, подскажет сведущему наблюдателю верные выводы".

*А. Конан Дойл*

Логической основой компьютера является алгебра логики, которая рассматривает логические операции над высказываниями. В XIX веке английский математик **Джордж Буль** разработал алгебраические методы выполнения операций над логическими значениями "истина" (true) и "ложь" (false).

**Алгебра логики** – это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

**Логическое высказывание** – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

*Пример.* «3 – простое число» является высказыванием, поскольку оно истинно. Не всякое предложение является логическим высказыванием.

*Пример.* предложение «Давайте пойдем в кино» не является высказыванием. Вопросительные и побудительные предложения высказываниями не являются.

Алгебра логики рассматривает любое высказывание только с одной точки зрения – является ли оно истинным или ложным. Слова и словосочетания «не», «и», «или», «если..., то», «тогда и только тогда» и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются **ЛОГИЧЕСКИМИ СВЯЗКАМИ**.

Высказывания, образованные из других высказываний с помощью логических связок, называются **составными** (сложными). Высказывания, которые не являются составными, называются **элементарными** (простыми).

*Пример.* высказывание «Число 6 делится на 2» - простое высказывание. Высказывание «Число 6 делится на 2, и число 6 делится на 3» - составное

высказывание, образованное из двух простых с помощью логической связки «и».

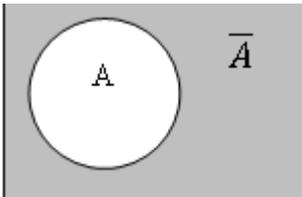
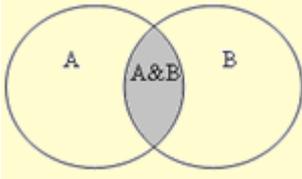
Истинность или ложность составных высказываний зависит от истинности или ложности элементарных высказываний, из которых они состоят.

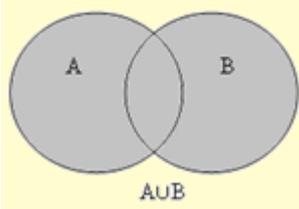
Чтобы обращаться к логическим высказываниям, им назначают имена.

**Пример.** Обозначим через А простое высказывание «число 6 делится на 2», а через В простое высказывание «число 6 делится на 3». Тогда составное высказывание «Число 6 делится на 2, и число 6 делится на 3» можно записать как «А и В». Здесь «и» – логическая связка, А, В – логические переменные, которые могут принимать только два значения – «истина» или «ложь», обозначаемые, соответственно, «1» и «0».

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение.

### Основные логические операции

Название операции	Форма записи	Таблица истинности	Круги Эйлера															
Операция «НЕ»; Логическое отрицание; Инверсия.	НЕ А; $\bar{A}$ ; $\neg A$ ; not А (Бейсик); !А (Си).	<table border="1"> <thead> <tr> <th>А</th> <th><math>\neg A</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> </tr> </tbody> </table>	А	$\neg A$	0	1	0	0										
А	$\neg A$																	
0	1																	
0	0																	
Операция «И»; Логическое умножение; Логическое произведение; Конъюнкция.	А И В; $A * B$ ; АВ; $A \wedge B$ ; $A \cap B$ ; А&В; А and В (Паскаль, Бейсик) А && В (Си).	<table border="1"> <thead> <tr> <th>А</th> <th>В</th> <th><math>A * B</math></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	А	В	$A * B$	0	0	0	0	1	0	1	0	0	1	1	1	
А	В	$A * B$																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

<p>Операция «ИЛИ»; Логическое сложение; Логическая сумма; Дизъюнкция.</p>	<p><b>A ИЛИ B;</b> <b>A+B;</b> <b>A∨B;</b> <b>A∪B;</b> <b>A or B (Паскаль, Бейсик)</b> <b>A  B (Си).</b></p>	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A+B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	A+B	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	A+B																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
<p>Операция «Если - то»; Логическое следование; Импликация.</p>	<p><b>A→B.</b></p>	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A→B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	A→B	0	0	1	0	1	1	1	0	0	1	1	1	
A	B	A→B																
0	0	1																
0	1	1																
1	0	0																
1	1	1																
<p>Равносильность; Эквиваленция.</p>	<p><b>A↔B.</b></p>	<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>A↔B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	A↔B	0	0	1	0	1	0	1	0	0	1	1	1	
A	B	A↔B																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

**НЕ** Операция, выражаемая словом «не», называется **отрицанием** и обозначается чертой над высказыванием (или знаком  $\neg$ ). Высказывание  $\neg A$  истинно, когда  $A$  ложно, и ложно, когда  $A$  истинно.

*Пример.* Пусть  $A$ =«Сегодня пасмурно», тогда  $\neg A$ =«Сегодня не пасмурно».

**И** Операция, выражаемая связкой «и», называется **конъюнкцией** (лат. conjunctio – соединение) или логическим умножением и обозначается точкой « $\cdot$ » (может также обозначаться знаками  $\wedge$  или  $\&$ ). Высказывание  $A \cdot B$  истинно тогда и только тогда, когда оба высказывания  $A$  и  $B$  истинны.

*Пример.* Высказывание «Число 6 делится на 2, и число 6 делится на 3» - истинно, а высказывание «Число 6 делится на 2, и число 6 больше 10» - ложно.

**ИЛИ** Операция, выражаемая связкой «или» (в неисключающем смысле этого слова), называется **дизъюнкцией** (лат. disjunctio – разделение) или логическим сложением и обозначается знаком  $\vee$  (или плюсом). Высказывание  $A \vee B$  ложно тогда и только тогда, когда оба высказывания  $A$  и  $B$  ложны.

**Пример:** Высказывание «Число 6 делится на 2 или число 6 больше 10» - истинно, а высказывание «Число 6 делится на 5 или число 6 больше 10» - ложно.

**ЕСЛИ ... ТО** Операция, выражаемая связками «если ..., то», «из ... следует», «... влечет ...», называется **импликацией** (лат. *implico* – тесно связаны) и обозначается знаком  $\rightarrow$ . Высказывание  $A \rightarrow B$  ложно тогда и только тогда, когда  $A$  истинно, а  $B$  ложно.

**Пример.** Высказывание «если студент сдал все экзамены на «отлично», то он получит стипендию». Очевидно, эту импликацию следует признать ложной лишь в том случае, когда студент сдал на «отлично» все экзамены, но стипендии не получил. В остальных случаях, когда не все экзамены сданы на «отлично» и стипендия получена (например, в силу того, что студент проживает в малообеспеченной семье) либо когда экзамены вообще не сданы и о стипендии не может быть и речи, импликацию можно признать истинной.

**РАВНОСИЛЬНО** Операция, выражаемая связками «тогда и только тогда», «необходимо и достаточно», «... равносильно ...», называется **эквиваленцией** или **двойной импликацией** и обозначается знаком  $\leftrightarrow$  или  $\sim$ . Высказывание  $A \leftrightarrow B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают.

**Пример:** Высказывание «Число является четным тогда и только тогда, когда оно делится без остатка на 2» является истинным, а высказывание «Число является нечетным тогда и только тогда, когда оно делится без остатка на 2» - ложно.

**ЛИБО ... ЛИБО** Операция, выражаемая связками «Либо ... либо», называется **исключающее ИЛИ** или **сложением по модулю 2** и обозначается XOR или  $\oplus$ . Высказывание  $A \oplus B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  не совпадают.

**Пример.** Высказывание «Число 6 либо нечетно либо делится без остатка на 2» является истинным, а высказывание «Либо число 6 четно либо число 6 делится на 3» – ложно, так как истинны оба высказывания входящие в него.

**Замечание.** Импликацию можно выразить через дизъюнкцию и отрицание:

$$A \rightarrow B = \neg A \vee B.$$

Эквиваленцию можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \leftrightarrow B = (\neg A \vee B) \wedge (\neg B \vee A).$$

Исключающее ИЛИ можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \text{ XOR } B = (\neg A \wedge B) \vee (\neg B \wedge A).$$

**Вывод.** Операций отрицания, дизъюнкции и конъюнкции достаточно, чтобы описывать и обрабатывать логические высказывания.

Порядок выполнения логических операций задается круглыми скобками. Но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания («не»), затем конъюнкция («и»), после конъюнкции – дизъюнкция («или») и исключающего или и в последнюю очередь – импликация и эквиваленция.

С помощью логических переменных и символов логических операций любое высказывание можно формализовать, то есть заменить логической формулой (логическим выражением).

**Логическая формула** - это символическая запись высказывания, состоящая из логических величин (констант или переменных), объединенных логическими операциями (связками).

**Логическая функция** - это функция логических переменных, которая может принимать только два значения: 0 или 1. В свою очередь, сама логическая переменная (аргумент логической функции) тоже может принимать только два значения: 0 или 1.

**Пример.**  $F(A, B) = A \& B \vee A$  – логическая функция двух переменных A и B.

Значения логической функции для разных сочетаний значений входных переменных – или, как это иначе называют, наборов входных переменных – обычно задаются специальной таблицей. Такая таблица называется **таблицей истинности**.

Приведем таблицу истинности основных логических операций

A	B	$\neg A$	$A \& B$	$A \vee B$
1	1	0	1	1
1	0	0	0	1
0	1	1	0	1
0	0	1	0	0

Опираясь на данные таблицы истинности основных логических операций можно составлять таблицы истинности для более сложных формул.

**Алгоритм построения таблиц истинности** для сложных выражений:

1. Определить количество строк:
  - количество строк =  $2^n$  + строка для заголовка,
  - n - количество простых высказываний.
2. Определить количество столбцов:

- количество столбцов = количество переменных + количество логических операций;
- определить количество переменных (простых выражений);
- определить количество логических операций и последовательность их выполнения.

**Пример 1.** Составить таблицу истинности для формулы И–НЕ, которую можно записать так:  $\neg(A \& B)$ .

1. Определить количество строк:

На входе два простых высказывания: А и В, поэтому  $n=2$  и количество строк  $=2^2+1=5$ .

2. Определить количество столбцов:

Выражение состоит из двух простых выражений (А и В) и двух логических операций (1 инверсия, 1 конъюнкция), т.е. количество столбцов таблицы истинности = 4.

3. Заполнить столбцы с учетом таблиц истинности логических операций

А	В	$A \& B$	$\neg(A \& B)$
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	1

Подобным образом можно составить таблицу истинности для формулы ИЛИ–НЕ, которую можно записать так:

$\neg(A \vee B)$ .

А	В	$A \vee B$	$\neg(A \vee B)$
1	1	1	0
1	0	1	0
0	1	1	0
0	0	0	1

**Пример 2.** Составить таблицу истинности логического выражения  $C = \neg A \& B \vee A \& \neg B$ .

**Решение:**

1. Определить количество строк:

На входе два простых высказывания: А и В, поэтому  $n=2$  и количество строк  $=2^2+1=5$ .

2. Определить количество столбцов:

Выражение состоит из двух простых выражений (А и В) и пяти логических операций (2 инверсии, 2 конъюнкции, 1 дизъюнкция), т.е. количество столбцов таблицы истинности = 7.

Сначала выполняются операции инверсии, затем конъюнкции, в последнюю очередь операция дизъюнкции.

3. Заполнить столбцы с учетом таблиц истинности логических операций

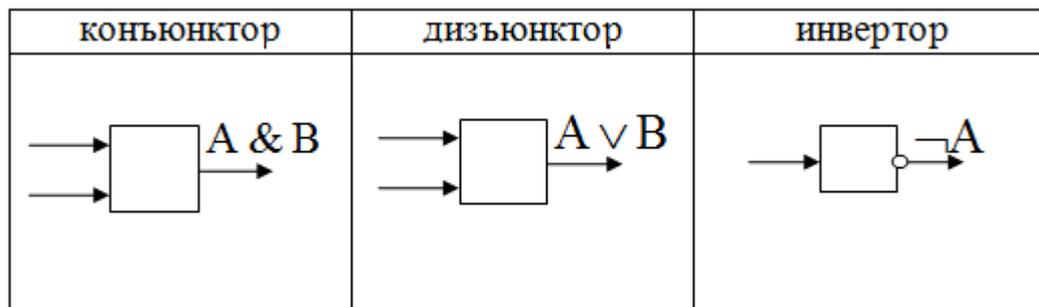
Таблица истинности для логической операции  $C = \neg A \& B \vee A \& \neg B$

A	B	$\neg A$	$\neg B$	$\neg A \& B$	$A \& \neg B$	C
1	1	0	0	0	0	0
1	0	0	1	0	1	1
0	1	1	0	1	0	1
0	0	1	1	0	0	0

Логические формулы можно также представлять с помощью языка **логических схем**.

Существует три базовых логических элемента, которые реализуют три основные логические операции:

- логический элемент «И» – логическое умножение – конъюнктор;
- логический элемент «ИЛИ» – логическое сложение – дизъюнктор;
- логический элемент «НЕ» – инверсию – инвертор.



Поскольку любая логическая операция может быть представлена в виде комбинации трех основных, любые устройства компьютера, производящие обработку или хранение информации, могут быть собраны из базовых логических элементов, как из “кирпичиков”.

Логические элементы компьютера оперируют с сигналами, представляющими собой электрические импульсы. Есть импульс – логический смысл сигнала – 1, нет импульса – 0. На входы логического элемента поступают сигналы-значения аргументов, на выходе появляется сигнал-значение функции.

Преобразование сигнала логическим элементом задается таблицей состояний, которая фактически является таблицей истинности, соответствующей логической функции, только представлена в форме

логических схем. В такой форме удобно изображать цепочки логических операций и производить их вычисления.

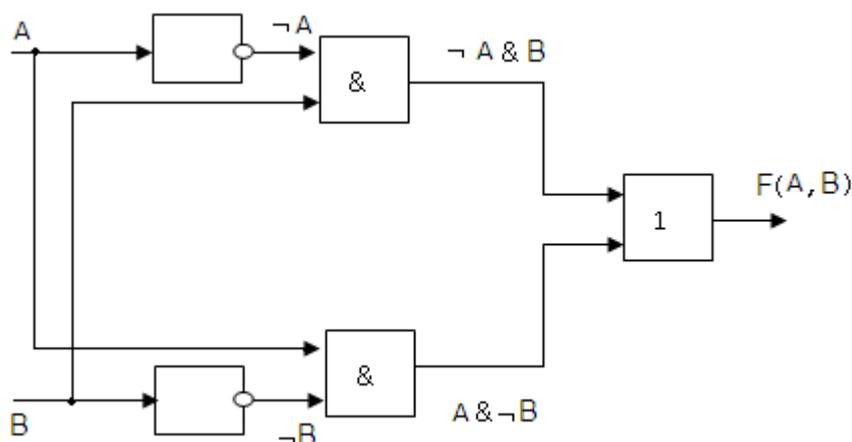
### Алгоритм построения логических схем

1. Определить число логических переменных.
2. Определить количество логических операций и их порядок.
3. Изобразить для каждой логической операции соответствующий ей логический элемент.
4. Соединить логические элементы в порядке выполнения логических операций.

**Пример.** По заданной логической функции  $F(A, B) = \neg A \& B \vee A \& \neg B$  построить логическую схему.

### Решение.

1. Число логических переменных = 2 (A и B).
2. Количество операций = 5 (2 инверсии, 2 конъюнкции, 1 дизъюнкция). Сначала выполняются операции инверсии, затем конъюнкции, в последнюю очередь операция дизъюнкции.
3. Схема будет содержать 2 инвертора, 2 конъюнктора и 1 дизъюнктор.
4. Построение надо начинать с логической операции, которая должна выполняться последней. В данном случае такой операцией является логическое сложение, следовательно, на выходе должен быть дизъюнктор. На него сигналы подаются с двух конъюнкторов, на которые, в свою очередь, подаются один входной сигнал нормальный и один инвертированный (с инверторов).



## 2.2. Вопросы для самопроверки по теме 2

2.3.

**Задание № 1.** Найдите среди заданных логических функций тождественно ложную:

1. A и не A или не A;
2. A и не B и не A;

3. A и не A или B;

4. A и не B или A.

Задание №2. Укажите, какой логической операции она соответствует приведенная таблица истинности:

A	B	F
1	1	1
1	0	1
0	1	1
0	0	0

Задание №3. Укажите, каким высказыванием является утверждение: "2+3=4".

Задание № 4. Найдите среди заданных логических функций тождественно истинную:

1. A и не A или B;

2. A или не B или не A;

3. A и не A или не A;

4. A и не B или A.

Задание № 5. Чтобы логическое выражение  $\overline{(A \vee B) \& (A \vee \overline{B})}$  было истинным, логическая переменная A должна принять значение

1. Истина

2. Ложь

3. B

4. -B

Задание № 6. Таблице истинности вида

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

соответствует какая логическая схема ...

Задание № 7. Логическая функция дизъюнкция имеет значение «истина», если:

1. обе переменные истинны;

2. обе переменные ложны;

3. хотя бы одна переменная истинна;

4. хотя бы одна переменная ложна.

### 3. ТЕХНИЧЕСКИЕ СРЕДСТВА РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

#### 3.1. Этапы развития вычислительной техники

Историю совершенствования механизмов, облегчающих вычисления, можно разделить на три основных этапа:

1) **Механический**: регистрируются механические перемещения элементов конструкции. Так как при этом можно предусмотреть любое количество различных состояний, конструкции этого этапа ориентированы на десятичную систему счисления. В истории развития этих механизмов можно выделить следующие этапы:

– *простейшие ручные приспособления* (период с IV тысячелетия до н.э). К ним относятся палочки, счёты абак: глиняная пластинка с желобами, в которых определённым образом раскладывались камешки, русские счёты: камешки нанизаны на проволоку;

– *вычислительные устройства*: арифмометры разных конструкций (с середины XVII века). Первый удобный для расчетов арифмометр создал Блез Паскаль в 1642 году. Его машина могла выполнять сложение и вычитание чисел с 6 – 8 разрядами и имела небольшие габариты. Следующий этап в принципиальном усовершенствовании арифмометров принадлежит Лейбницу. В 1673 году он представил машину, которая могла выполнять четыре арифметических действия;

– *автоматизация вычислений* – механические устройства, работающие по заданной программе. Идея разделения информации на команды и данные принадлежит Чарльзу Бэббиджу, который в 1822 году представил машину, которая могла рассчитывать таблицы не очень сложных функций.

В механических арифмометрах использовался принцип работы часового механизма: система взаимосвязанных зубчатых колес разного диаметра, в которой поворот каждого колеса на один зубчик соответствовал изменению на единицу определенного разряда числа.

2) **Электромеханический** – в счетных устройствах используются электромагнитные реле (первая половина XX века). Первая машина такого типа была построена немецким инженером Конрадом Цузе в 1941 году. В 1943 году появились машины Марк-1, затем Марк-2, созданные американцем Говардом Эйкеном. Эти машины выполняли арифметические операции с 23-значными десятичными числами и работали гораздо быстрее механических.

3) **Электронный** – регистрируются не механические смещения, а состояния элементов конструкции. При этом оказалось удобнее всего использовать не десятичную, а двоичную систему счисления (включено/выключено, заряжено/разряжено, есть контакт/нет контакта). Первая машина такого типа, ENIAC (Electronic Numerical Integrator And Computer), была создана в США под руководством группы специалистов Говарда Эйкана, Дж. Моучли, П. Эккерта и введена в эксплуатацию 15.02.1945 г.

Создание ЭНИАК является точкой отсчета пути, по которому пошло развитие ЭВМ.

В СССР первая ЭВМ была создана 1951 году под руководством академика С.А. Лебедева и называлась МЭСМ (малая электронная счетная машина). Его машины БЭСМ – 1, БЭСМ-3М, БЭСМ-4, М- 220 были признаны лучшими в мире.

### 3.2. Классификация и состав ЭВМ

#### 1. По производительности:

Среди всего множества современных ЭВМ можно выделить основные классы

- СуперЭВМ;
- Большие вычислительные комплексы (Мейнфреймы);
- Мини-ЭВМ;
- Персональные ЭВМ (ПЭВМ).

#### ***СуперЭВМ***

СуперЭВМ предназначены для решения сверхсложных задач в военном деле, экономике, космонавтике, метеорологии и др.

Это очень сложные и дорогие машины. Выпускаются небольшими партиями для конкретной задачи или конкретного заказчика. Машин такого уровня около 500 в мире. Штат обслуживания большой ЭВМ составляет многих десятков человек. Производительность – свыше 100 млн. операций в секунду.

Лучшие ПЭВМ по производительности примерно в 100 тыс. раз слабее суперЭВМ.

#### ***Большие вычислительные комплексы.***

В 90-х годах они были очень распространены в СССР и во все мире. В СССР это были в основном различные модификации серии ЕС. Ориентировочные данные подобных ЭВМ: быстродействие до 50 млн. опер./сек; объем ОЗУ до 8 Мбайт; занимаемая площадь от 50 до 200 м<sup>2</sup>.

БВК (получившие название «Мейнфреймы») выпускаются и в настоящее время, но современные технологии позволили резко уменьшить их габариты: массу до 100 кг; занимаемую площадь до 12 м<sup>2</sup>.

Область их применения – хранение, поиск и обработка больших массивов данных, построение трехмерной анимационной графики, создание рекламных роликов, выполняют роль узлов глобальной сети, там, где требуется исключительная надежность работы.

#### ***Мини-ЭВМ.***

Ранее они использовались в небольших организациях для решения сравнительно несложных задач. В СССР были распространены мини-ЭВМ серии СМ. Примерные данные их: быстродействие до 1,5 млн. Опер/сек.; занимаемая площадь до 30 м<sup>2</sup>.

Современные мини-ЭВМ, благодаря достижениям микроэлектроники, по размерам сравнялись с ПЭВМ, имея огромное преимущество над последними в производительности и надежности.

Они находят применение, например, в банковской сфере, в качестве серверов (центральных ЭВМ) высоконадежных локальных вычислительных сетей с числом рабочих станций до 300.

### ***Персональные ЭВМ (микро-эвм).***

Эта категория получила бурное развитие в течение последних 25 лет.

На компьютерном рынке сложилось следующее разделение конфигураций персональных компьютеров.

### **Рабочая станция (WorkStation).**

Представляет собой мощный компьютер, основанный обычно на двухпроцессорной платформе, оснащенный максимальным объемом быстрой оперативной памяти, массивом жестких дисков и часто включенный в локальную сеть предприятия. В зависимости от решаемых задач рабочие станции бывают графическими, для научных расчетов или иного назначения.

### **Настольный компьютер (Desktop).**

Предусматривает самый обширный спектр возможных конфигураций.

Принято классифицировать настольные компьютеры по назначению и по производительности.

По *назначению* компьютеры подразделяют на офисные, домашние, игровые, дизайнерские.

По *производительности* различают компьютеры начального уровня (EasyPC), среднего уровня (Mainstream), высшего класса (HighEnd).

*Офисный* компьютер обычно ориентирован на работу с программами офисного класса, может подключаться к локальной сети и не отличается высокой производительностью. Главное требование к нему – надежность.

*Домашний* компьютер используют для развлечений и выполнения не слишком сложных учебных (рабочих) заданий. Мультимедийная направленность выражается в оснащении его процессором и видеокартой среднего класса, приводом DVD, качественным монитором и комплексом хорошей акустики. Дополнительным оборудованием являются сканер, струйный фотопринтер, Web-камера, подключение к Интернету через модем или сетевую плату.

*Игровой* компьютер требует наличия самой мощной графической подсистемы. Поэтому главным его элементом является графическая карта и адекватный потребностям процессор при достаточном объеме оперативной памяти. Дополнительно его комплектуют джойстиком, рулем (штурвалом), педалями, устройствами виртуальной реальности (шлемы, очки, перчатки).

*Дизайнерский* компьютер предназначен для выполнения сложных графических работ и обработке видео в режиме реального времени. По сути это рабочая станция начального уровня, в достаточно компактном исполнении.

### **Ноутбук (Notebook)**

Является переносным персональным компьютером. Он имеет компактные габариты и встроенные аккумуляторы, позволяющие работать без сетевого напряжения.

### **Настольный ноутбук (DeskNote).**

Этот класс компьютеров возник и развился в 2002 году. Его отличие от ноутбуков заключается в отсутствии аккумуляторов, использовании процессоров для обычных настольных ПК.

### **Планшетный ПК (TabletPC).**

Характеризуется наличием отдельного сенсорного дисплея с возможностью рукописного ввода и специального электронного пера. Некоторые модели комплектуются клавиатурой, трекболом, приводом CD-ROM, жестким диском.

### **Карманный ПК (Personal Digital Assistant, PDA).**

Это самый маленький ПК. Он не имеет внешней памяти на магнитных дисках, она заменена на энергозависимую электронную память. Эта память может перезаписываться при помощи линии связи с настольным компьютером. Карманный компьютер можно использовать как словарь-переводчик или записную книжку.

## **2. По типу обрабатываемых сигналов (см. выше):**

- ЭЦВМ (цифровые).
- АВМ (аналоговые).
- Гибридные (смешанные).

## **3. По элементной базе**

В вычислительной технике существует своеобразная периодизация развития электронных вычислительных машин. Вся электронно-вычислительную технику принято делить на поколения. ЭВМ относят к тому либо иному поколению в зависимости от типа главных используемых в ней частей либо от технологии их производства. От элементной базы зависит мощность компьютера, что в свою очередь привело к изменениям в архитектуре ЭВМ, расширению круга ее задач, к изменению способа взаимодействия пользователя и компьютера. Ясно, что границы поколений в смысле времени сильно размыты, так как в одно и то же время практически выпускались ЭВМ разных типов; для отдельной же машины вопрос о её принадлежности к тому либо иному поколению решается довольно просто.

### 3.3. Поколения ЭВМ

*Под поколением понимают все типы и модели ЭВМ, разработанные различными конструкторскими коллективами, но построенные на одних и тех же научных и технических принципах*

По элементной базе выделяют 5 поколений ЭВМ (периоды указаны условно):

- *первое поколение* – на электровакуумных лампах (1945– 1955 г.г.);
- *второе поколение* – на транзисторах (1955-1965 г.г.);
- *третье поколение* – на микросхемах. Разрабатываются семейства машин с единой архитектурой, что приводит к программной совместимости, т. е. при появлении новой марки ЭВМ отпала необходимость переписывать заново все программы, которые были разработаны для предыдущей марки (1965–1970 г.г.);
- *четвёртое поколение* – на интегральных схемах. Это существенно увеличило скорость работы, уменьшило энергоёмкость, стоимость и габариты ЭВМ. Происходит переход к персональным ЭВМ. Создаются многопроцессорные и многомашинные комплексы (с 1970 г.);
- *пятое поколение* – суперкомпьютеры на больших интегральных схемах. Используются магнитные, лазерные, голографические принципы различения состояний. Машины этого поколения ориентированы на логическое программирование (обслуживание экспертных систем, плохо формализованных задач).

### 3.4. Архитектура персонального компьютера (ПК)

**Компьютер** (от англ. computer – вычислитель) – это программируемое электронное устройство, предназначенное для накопления, обработки и передачи информации.

Существует два основных класса компьютеров:

- **цифровые компьютеры**, обрабатывающие данные в виде [двоичных кодов](#);
- **аналоговые компьютеры**, обрабатывающие непрерывно меняющиеся физические величины (электрическое напряжение, время и т.д.), которые являются аналогами вычисляемых величин.

Поскольку в настоящее время подавляющее большинство компьютеров являются цифровыми, далее будем рассматривать только этот класс компьютеров и слово "компьютер" употреблять в значении "цифровой компьютер".

Разнообразие современных компьютеров очень велико. Но их структуры основаны на **общих логических принципах**, позволяющих выделить в любом компьютере следующие **главные устройства**

**Архитектура компьютера** – это совокупность аппаратных и программных средств ПК, а также система взаимодействия их, обеспечивающая функционирование ПК.

*Классическая архитектура ЭВМ и принципы фон Неймана.*

**Джон фон Нейман** (американский математик) заложил основы учения об архитектуре.

**В 1946 г. Нейман со своими коллегами Г. Голдстайном и А. Берксом изложил принципы построения вычислительной машины:**

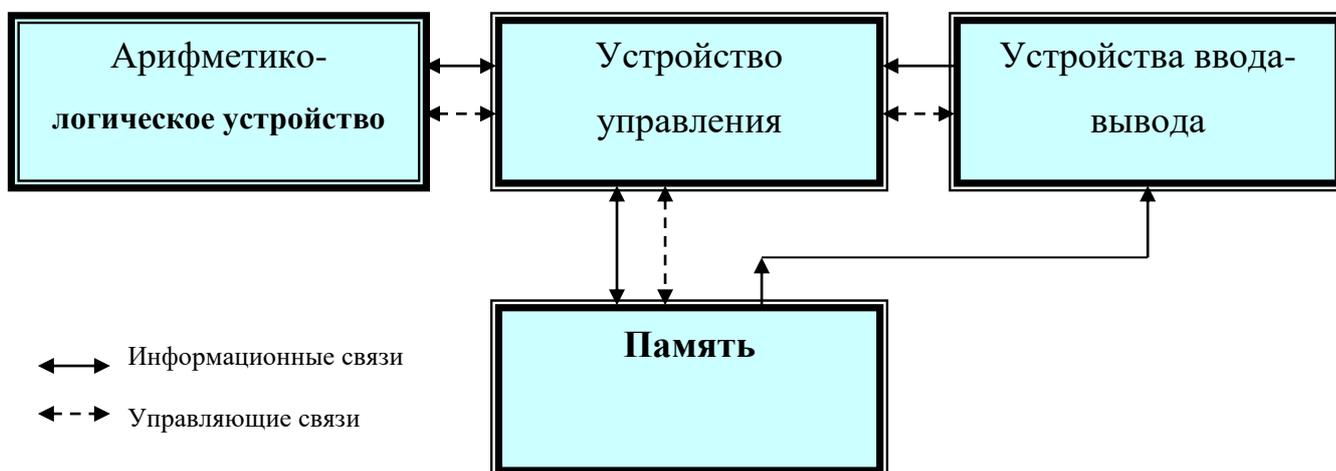
1. Принцип программного управления. Из него следует, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности, без вмешательства человека.

2. Принцип однородности памяти. Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти — число, текст или команда. Над командами можно выполнять такие же действия, как и над данными. Это открывает целый ряд возможностей. Например, программа в процессе своего выполнения также может подвергаться переработке, что позволяет задавать в самой программе правила получения некоторых ее частей (так в программе организуется выполнение циклов и подпрограмм). Более того, команды одной программы могут быть получены как результаты исполнения другой программы. На этом принципе основаны методы [трансляции](#) — перевода текста программы с языка программирования высокого уровня на язык конкретной машины.

3. Принцип адресности. Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена

областям памяти, так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

Согласно идеям фон Неймана вычислительная машина должна состоять из следующих основных компонентов:



- *Арифметическо-логическое устройство* – устройство для выполнения арифметических и логических операций
- *Устройство управления* отвечает за процесс выполнения программ
- *Оперативная память* – устройство для запоминания информации
- *Устройства ввода-вывода* используются для ввода-вывода информации

Основные особенности архитектуры современных ПК - **открытость и модульность**.

Открытость означает возможность замены отдельных компонентов ПК их более совершенными версиями, а также возможность подключения новых устройств к компьютеру с целью расширения его возможностей.

Все компоненты компьютера оформлены в виде законченных конструкций – модулей, имеющих стандартные размеры и стандартные средства соединения с ЭВМ. Они не связаны жестко, в единое неразъемное устройство: предусмотрена возможность быстрого подсоединения и отсоединения любого из них к ПК. Кроме того, в любой ЭВМ подобного типа используется стандартный набор основных модулей, при любой её модификации.

### 3.5. Базовая аппаратная конфигурация ПК

**Персональный компьютер** - универсальная ЭВМ, предназначенная для индивидуального пользования. Его конфигурацию (состав оборудования) можно гибко изменять по мере необходимости. Тем не менее, существует

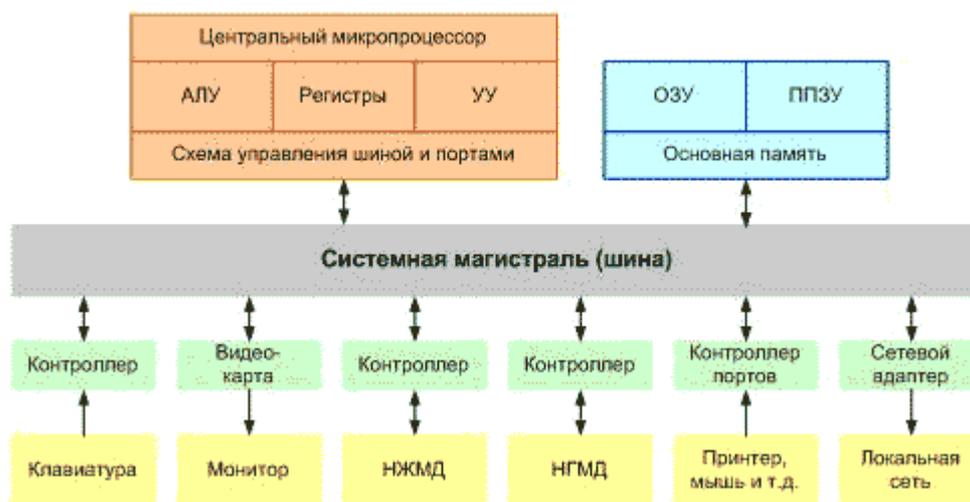
понятие **базовой конфигурации ПК**, которую считают типовой. Понятие **базовой конфигурации** может меняться, в настоящее время в нее входят **4 устройства**: системный блок, монитор, клавиатура, мышь.

1) **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 – **внешними (периферийные)** – это дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных.

**Внутренние** устройства системного блока:

1. **Материнская плата**
2. **Жесткий диск**
3. **Дисководы гибких дисков**
4. **Дисковод компакт-дисков: CD-ROM, CD-R, CD-RW, DVD**
5. **Адаптеры: видеоадаптер, звуковая плата, сетевая плата и др.**
6. **Порты ввода-вывода**
7. **Блок питания**

Структурная схема ПК



1. **Материнская плата** - основная плата ПК, которая определяет его конфигурацию. На ней размещаются:

- **Процессор (CPU)** – основная микросхема, которая производит все арифметические и логические операции, осуществляет управление всем процессом решения задачи по заданной программе, т.е. является главным компонентом компьютера. Процессоры выполнены в виде одной микросхемы и поэтому называются также **микропроцессорами**. Тип ПК определяется типом процессора.

Главные характеристики процессора:

– **Разрядность** процессора показывает, сколько двоичных разрядов(бит) информации обрабатывается за один такт (время выполнения одной элементарной операции) – 8, 16, 32 (в старых моделях) или 64.

– **Тактовая частота** показывает, сколько элементарных операций (тактов) процессор выполняет в течение одной секунды. Чем выше тактовая частота процессора, тем быстрее он работает. Единица измерения тактовой частоты - **мегагерц (МГц)**.

Первые **16-ти разрядные** процессоры (например, Intel-8086) работали с **тактовой частотой** не выше 4,7МГц.

- **Оперативная память (оперативное запоминающее устройство, ОЗУ или RAM)** – набор микросхем, предназначенных для временного хранения данных, когда компьютер включен. Объем памяти в современном ПК от 128 Мбайт и выше.

- **Постоянная память (постоянное запоминающее устройство, ПЗУ или ROM)** – микросхема, предназначенная для длительного хранения данных, в том числе и когда компьютер выключен. Комплект программ, находящихся в ПЗУ, образует **базовую систему ввода-вывода (BIOS)**. Основное назначение программ этого пакета состоит в том, чтобы проверить состав и работоспособность компьютерной системы и обеспечить взаимодействие с клавиатурой, монитором, жестким диском и дисководом гибких дисков.

- **Энергонезависимая память CMOS** её содержимое не стирается во время выключения ПК (отличие от оперативной памяти) и в неё можно вносить и изменять данные самостоятельно, в соответствии с тем, какое оборудование входит в состав системы.

- **Быстрая память (сверхоперативная, КЭШ-память)** используется для ускорения операций в памяти ПК. В КЭШ-память записывается часть информации из ОЗУ, с которой процессор работает в данный момент времени (буферная область). Создана для того, чтобы уменьшить количество обращений к оперативной памяти.

- **Системная шина** – система объединенных проводников для передачи информации между подключёнными к ней устройствами ПК. По шине передаётся информация трёх типов: адреса, адреса данных, команды.

**2. Жесткий диск (винчестер)** - основное устройство для долговременного хранения больших объемов информации, представляет собой группу соосных дисков, имеющих магнитное покрытие и вращающихся с высокой скоростью.

Основными параметрами жестких дисков являются **ёмкость** и **производительность**. Ёмкость дисков зависит от технологии их изготовления. В настоящее время большинство производителей жестких дисков используют изобретенную компанией IBM технологию с использованием **гигантского магниторезистивного эффекта**. В настоящее время на пластину может приходиться 40 и более Гбайт и развитие продолжается. Объем винчестера современной ПК должен быть не менее 1 Гбайта.

**Производительность** жестких дисков меньше зависит от технологии их изготовления. Сегодня все жесткие диски имеют очень высокий

показатель скорости внутренней передачи данных (до 30-60 Мбайт/с), и поэтому их производительность зависит от характеристик интерфейса, с помощью которого они связаны с материнской платой.

**3. Дисководы гибких магнитных дисках** (накопители на гибких магнитных дисках (НГМД) представляют собой ЗУ, в которых носителями информации являются сменные магнитные диски (дискеты), которые вставляются в специальный накопитель - **ДИСКОВОД**.

**Назначение:** обмен информацией между компьютерами: сохранение данных вне компьютера, создание архивов данных, и в частности архивных копий текстов и программ, записанных на винчестере.

В настоящее время используются обычные гибкие диски и флоппические гибкие диски.

#### **4. Дисководы для работы с лазерным диском**

Эти системы служат для чтения информации с компакт-дисков различного вида и пользуются в настоящее время исключительным спросом.

Рассмотрим виды этих дисков:

- **CD-ROM** – компакт-диск только для чтения.
- **CD-R**- диски. Запись на **CD-R**-диск возможна только 1 раз, производится пользователем с помощью компактного и недорогого записывающего дисковода.
- **CD-RW**-диски. Обеспечивают возможность многократной перезаписи информации на диск (до 1000 циклов) пользователем с помощью специального записывающего дисковода.
- **DVD**-диски. Это так называемые цифровые диски. Основное отличие их от прочих лазерных дисков – значительно более высокая плотность записи информации.

**5. Адаптеры** - это печатные платы с микросхемами, которые преобразуют различные формы представления информации для организации различных устройств компьютера. Например: видеоадаптер (видеокарта), звуковая плата, сетевая плата и др.

**6. Порты ввода-вывода**- разъемы, с помощью которых к системному блоку подключаются периферийные устройства (Принтер, «мышь» и т.д.). Порты общего назначения бывают двух типов:

- параллельные (обозначаемые LPT1 – LPT4) – обычно 25 контактов;
- последовательные (обозначаемые COM1 – COM3) – обычно 9 контактов, но возможно и 25.

К параллельному порту подключается, например, принтер, к последовательному – мышь.

Параллельные порты выполняют ввод-вывод данных с большой скоростью, но требуют и большого числа проводов.

Не так давно появился новый быстродействующий вид порта – USB. Он позволяет подключать до 256 устройств, прерывать работу с ПК в активном режиме.

Итак, основными характеристиками материнской платы являются:

1. Максимально допустимая частота процессора;
2. Число разъемов для плат расширения;
3. Тип системной шины.

8. **Блок питания** преобразует переменный ток сети электропитания в постоянный ток низкого напряжения. Устройства компьютера используют напряжение 12,5 и 3,5 Вольт.

### **Память ПК**

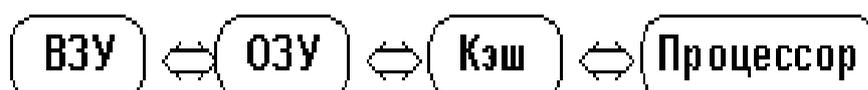
Современные компьютеры имеют много запоминающих устройств (ЗУ), которые сильно отличаются между собой по назначению, временным характеристикам, объему хранимой информации. Различают два вида памяти:

1. Внутренняя память
2. Внешняя память

В состав **внутренней памяти** входят:

- *оперативная память,*
- *кэш-память*
- *специальная память.*

**Внешняя память (ВЗУ)** предназначена для длительного хранения программ и данных, и целостность её содержимого не зависит от того, включен или выключен компьютер. В отличие от оперативной памяти, **внешняя память не имеет прямой связи с процессором**. Информация от ВЗУ к процессору и наоборот циркулирует примерно по следующей цепочке:



### 3.6. Периферийные устройства

Внешние (периферийные) устройства персонального компьютера составляют важнейшую часть любого вычислительного комплекса. Стоимость внешних устройств, в среднем, составляет около 80-85% стоимости нашего комплекса. Внешние устройства обеспечивают взаимодействие компьютера с окружающей средой — пользователями, объектами управления и другими компьютерами.

Внешние устройства подключаются к компьютеру через специальные разъемы-порты ввода-вывода.

**К внешним устройствам относятся:**

- **устройства ввода информации;**
- **устройства вывода информации;**
- **диалоговые средства пользователя;**
- **средства связи и телекоммуникации.**

**Устройства ввода** преобразуют информацию из естественной формы, доступной органам чувств человека, в двоичную форму:

- **клавиатура** — устройство для ручного ввода в компьютер числовой, текстовой и управляющей информации;
- **графические планшеты (дигитайзеры)** — для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера); при перемещении пера автоматически выполняется считывание координат его местоположения и ввод этих координат в компьютер;
- **сканеры (читающие автоматы)** — для автоматического считывания с бумажных носителей и ввода в компьютер машинописных текстов, графиков, рисунков, чертежей;
- **устройства указания (графические манипуляторы)** — для ввода графической информации на экран монитора путем управления движением курсора по экрану с последующим кодированием координат курсора и вводом их в компьютер (джойстик, мышь, трекбол, световое перо);
- **сенсорные экраны** — для ввода отдельных элементов изображения, программ или команд с полиэкрана дисплея в компьютер);
- **цифровые видеокамеры и фотоаппараты;**
- **микрофон.**

**Устройства вывода** преобразуют информацию из двоичной формы в естественную форму, доступную органам чувств человека:

- **мониторы** – устройства вывода информации;
- **принтеры** — печатающие устройства для вывода информации на бумажный носитель;
- **графопостроители (плоттеры)** — для вывода графической информации на бумажный носитель;
- информация может также воспроизводиться в виде звуков с помощью **акустических колонок**

- **телефонов,**
- **адаптер:** устройство для соединения блоков компьютера с разными способами представления информации;

Основные виды принтеров:

- **матричные** — изображение формируется из точек, печать которых осуществляются тонкими иглами, ударяющими бумагу через красящую ленту. Знаки в строке печатаются последовательно. Количество игловок в печатающей головке определяет качество печати. Недорогие принтеры имеют 9 игловок. Более совершенные матричные принтеры имеют 18 и 24 иглы;

- **струйные** — в печатающей головке имеются тонкие трубочки — сопла, через которые на бумагу выбрасываются мельчайшие капельки чернил. Матрица печатающей головки обычно содержит от 12 до 64 сопел. В настоящее время струйные принтеры обеспечивают разрешающую способность до 50 точек на миллиметр и скорость печати до 500 знаков в секунду при отличном качестве печати, приближающемся к качеству лазерной печати. Струйные принтеры выполняют и цветную печать, но разрешающая способность при этом уменьшается примерно вдвое;

- **лазерные** — применяется электрографический способ формирования изображений. Лазер служит для создания сверхтонкого светового луча, вычерчивающего на Поверхности предварительно заряженного светочувствительного барабана контуры невидимого точечного электронного изображения. После проявления электронного Воображения порошком красителя (тонера), налипающей на разряженные участки, выполняется печать — перенос тонера с барабана на бумагу и закрепление изображения на бумаге разогревом тонера до его расплавления. Лазерные принтеры обеспечивают наиболее высококачественную печать с высоким быстродействием. Широко используются цветные лазерные принтеры.

К *диалоговым средствам пользователя* относятся:

- **видеотерминалы** (мониторы) — устройства для отображения вводимой и выводимой информации. Видеотерминал состоит из видеомонитора (дисплея) и видеоконтроллера (видеоадаптера).

- **устройства речевого ввода-вывода** информации. К ним относятся различные микрофонные акустические системы, а также различные синтезаторы звука, выполняющие преобразование цифровых кодов в буквы и слова, воспроизводимые через динамики или звуковые колонки, подсоединенные к компьютеру.

*Средства связи и телекоммуникации* используются для подключения компьютера к каналам связи, другим компьютерам и компьютерным сетям. К этой группе прежде всего относятся сетевые адаптеры. В качестве сетевого адаптера чаще всего используются модемы (модулятор-демодулятор).

Многие из названных выше устройств относятся к условно выделенной группе — **средствам мультимедиа**.

**Средства мультимедиа** — это комплекс аппаратных и программных средств, позволяющих человеку общаться компьютером, используя самые разные естественные для себя среды: звук, видео, графику, тексты, анимацию и др. К средствам мультимедиа относятся:

- устройства речевого ввода и вывода информации;
- микрофоны и видеокамеры, акустические и видеовоспроизводящие системы с усилителями, звуковыми колонками, большими видеоэкранами;
- звуковые и видеоплаты, платы видеозахвата, снимающие изображение с видеомэгнитофона или видеокамеры и вводящие его в компьютер;
- сканеры;
- внешние запоминающие устройства большой емкости на оптических дисках, часто используемые для записи звуковой и видеоинформации.

### **3.7. Вопросы для самопроверки по теме 3**

*Задание № 1.* Разрядностью микропроцессора является:

1. количество бит, обрабатываемых микропроцессором за один такт работы;
2. число команд, которое микропроцессор может исполнить за единицу времени;
3. количество регистров;
4. размер кэш-памяти.

*Задание № 2.* В принципы работы компьютера, сформулированные фон Нейманом, входят:

1. принцип однородности памяти;
2. принцип адресности;
3. принцип программного управления;
4. принцип детерминированности.

*Задание № 3.* Укажите, к какому поколению относятся персональные компьютеры.

*Задание № 4.* Укажите наиболее важные характеристики жидкокристаллического монитора:

1. цвет фона окна;
2. физический размер экрана;
3. объём хранимых данных; скорость обработки информации;
4. угол обзора.

*Задание № 5.* BIOS (basic input/output system) – это:

1. блок питания процессора;
2. программа загрузки пользовательских файлов;
3. биологическая операционная система;
4. набор программ, выполняющих автоматическое тестирование устройств после включения питания компьютера и загрузку операционной системы в оперативную память.

*Задание № 6.* По реализации пользовательского интерфейса операционные системы делятся на:

1. однопользовательские и многопользовательские;
2. однозадачные и многозадачные;
3. аппаратные и программные;
4. графические и неграфические.

*Задание № 7.* На каких видах внешней памяти невозможно случайно стереть информацию?

*Задание № 8.* Укажите устройство хранения данных, которое работает только при включенном питании:

1. ПЗУ;
2. гибкий магнитный диск.
3. ОЗУ;
4. жёсткий диск.

*Задание № 9.* Если размер кластера 512 байт, а размер файла 756 байт, то файл займет на диске:

1. один кластер;
2. полтора кластера;
3. два кластера;
4. три кластера.

*Задание № 10.* Укажите устройства, размещаемые на материнской плате ПК.

*Задание № 11.* По какому признаку классифицируются принтеры?

*Задание № 12.* Что обозначает термин «адаптер»?

*Задание № 13.* Укажите, какие из названий относятся к устройствам вывода данных:

1. плоттер;
2. процессор;
3. блок питания;
4. монитор;
5. сканер.

*Задание № 14.* Укажите устройства, которые не входят в состав внутренней памяти современного компьютера.

## **4. ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ**

### **4.1. Программное обеспечение персонального компьютера**

Ни компьютер в целом, ни его составные части не способны сами по себе обрабатывать информацию. Управляют работой компьютера программы, которые имеют различные функции и назначение.

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

**Команда** - это описание операции, которую должен выполнить компьютер. Как правило, у команды есть свой код (условное обозначение), исходные данные (операнды) и результат.

**Программное обеспечение (ПО) ПК (Software)** - совокупность программ, необходимых для обработки или передачи различных данных, предназначенных для многократного использования и применения разными пользователями.

Программное обеспечение условно можно разделить на 3 класса: Системное ПО, Прикладное ПО и Инструментальное ПО.

Системное ПО, в свою очередь, состоит из Базового ПО и Сервисного ПО.

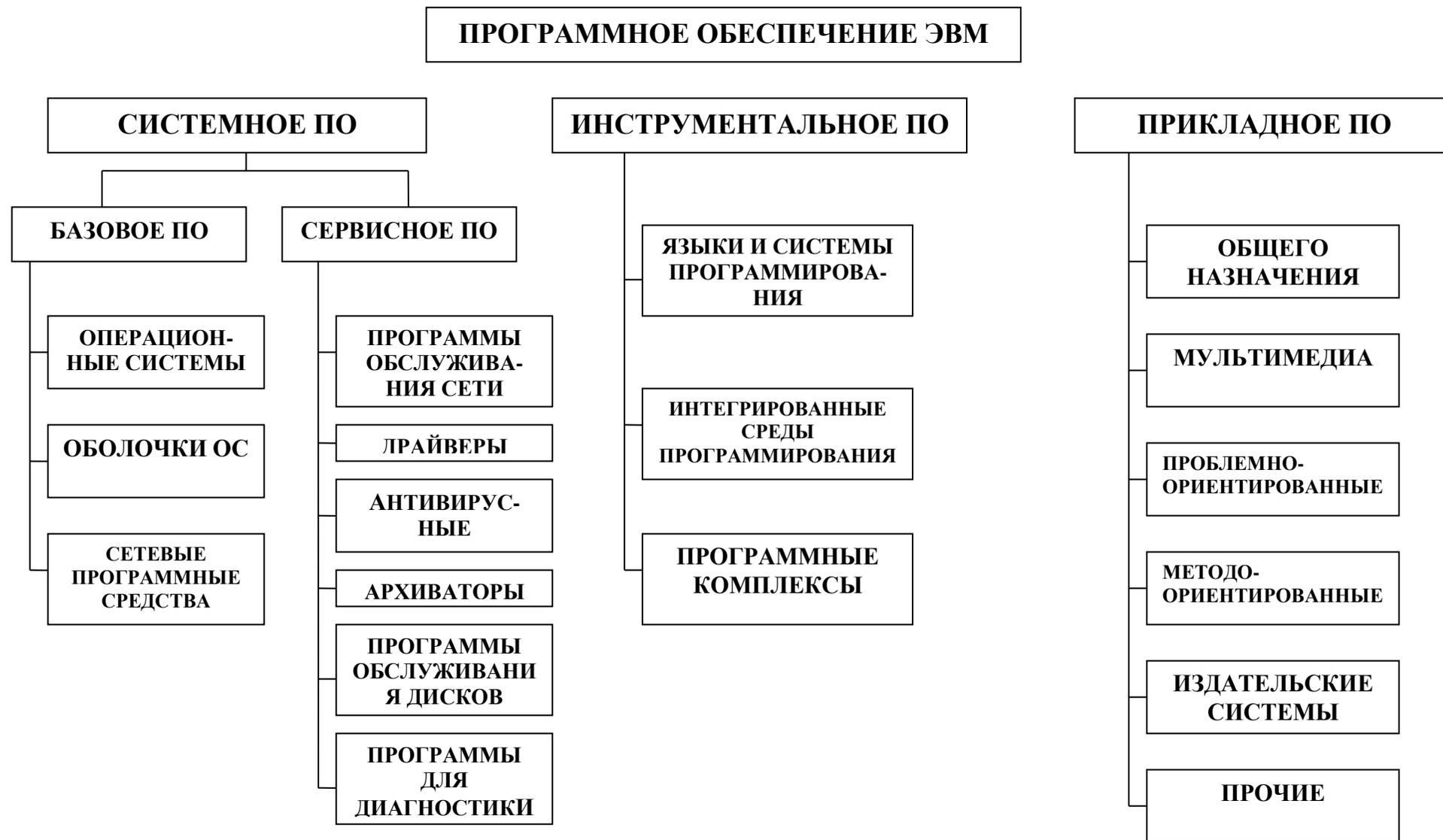


Рис. Структура программного обеспечения

**Базовое ПО** обеспечивает работоспособность компьютера.

**1. Операционная система** – это комплекс программ, предназначенных для наиболее эффективного использования всех средств ЭВМ в процессе решения задачи, для повышения удобства работы с ней.

Операционная система поставляется вместе с ЭВМ. Работа современной ЭВМ без неё невозможна. Операционная система предназначена:

1. Для запуска и нормальной работы компьютера;
2. Для функционирования других программ на компьютере;
3. Для организации связи, общения пользователя с ПК в целом и с отдельными её устройствами (устройствами печати, внешней памяти и т.д.);
4. Для диагностики и контроля работоспособности всех блоков ПК в процессе выполнения программы;
5. Для выполнения других вспомогательных технологических процессов.

Для каждого типа ПЭВМ используется своя ОС. На ПК одного типа могут применяться несколько видов ОС, различающихся своими возможностями, степенью общения с пользователем, способами реализации своих функций и т.д.

В настоящее время разработано большое количество ОС, различающихся по возможностям их функционирования. Широкое применение нашли следующие ОС: Windows, Linux, Mac OS, NetWare, OS/2, Solaris, QN

**2. Оболочка ОС** – это программа (комплекс программ), упрощающая работу с основной программой, обеспечивающая более наглядный и удобный способ общения с компьютером, чем средства ОС.

Наиболее популярными программами-оболочками для **MS-DOS** являются **NortonCommander** и **DOSNavigator**, для Windows – **WindowsCommander, FARManager**.

Например, работать с операционной системой **MS-DOS** достаточно сложно потому, что эта система управляется с помощью команд и эти команды пользователь должен помнить, кроме того, их нужно правильно набирать.

Разработано много оболочек над данной операционной системой, позволяющих упростить управление системой. В первую очередь это знаменитая оболочка (команды Нортон).

**Возможности Norton Commander:**

- 1) наглядно изображать содержание каталогов на дисках;
- 2) изображать дерево каталогов на диске, возможность перехода в нужный каталог, создавать, переименовывать и удалять каталоги.
- 3) копировать, переименовывать, пересылать и удалять файлы;
- 4) просматривать текстовые файлы, сделанные с помощью различных редакторов текстов, архивные файлы, графические файлы, базы данных;

- 5) редактировать текстовые файлы;
- 6) выполнять любые команды DOS;
- 7) изменять атрибуты файлов;

и многое другое.

**FARManager** (русская разработка, созданная русским программистом Евгением Рошалем) – это работающая в текстовом режиме программа управления файлами для Windows 95, 98, NT, 2000, XP, которая обеспечивает обработку с длинными именами и имеет обширный набор дополнительных функций.

**3. Сетевые ОС** обеспечивают работу компьютера в сети и поддерживают все сетевые службы - электронную почту, обмен файлами, доступ к сайтам, общение между клиентами через Интернет и пр. (Solaris, Novell NetWare, Microsoft Windows NT).

**Сервисное ПО** – это программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

Это набор сервисных, дополнительно устанавливаемых программ, которые можно классифицировать по функциональному признаку следующим образом:

1. **Драйверы** – это специальные программы, обеспечивающие взаимодействие ОС с аппаратными устройствами (драйверы видеокарты, графического ускорителя, CD-ROM, клавиатуры, мыши, модема, сетевой карты и т.д.). При включении компьютера производится загрузка драйверов в оперативную память. Пользователь имеет возможность вручную установить или переустановить драйверы.

2. **Программы диагностики** работоспособности компьютера позволяют проверить конфигурацию ПК, выявить дефекты дисков и предотвратить потерю данных, хранящихся на дисках.

3. **Антивирусные программы** – это программы, обеспечивающие защиту компьютера, обнаружение компьютерных вирусов и восстановление зараженных файлов.

4. **Программы обслуживания дисков**, обеспечивающие проверку качества поверхности магнитного диска, контроль сохранности файловой системы на логическом и физической уровнях, сжатие дисков, создание страховых копий дисков, резервирование данных на внешних носителях и др.

5. **Программы архивирования данных**, которые обеспечивают процесс сжатия информации в файлах с целью уменьшения объема памяти для её хранения.

6. **Программы обслуживания сети.**

Эти программы часто называются **утилитами** (к антивирусным средствам этот термин обычно не применяется).

Наибольшее распространение сегодня имеют комплекты утилит: Norton Utilities - фирма Symantec; Checkit PRO Deluxe 2.0 - фирма Touch

Stone; PC Tools for Windows 2.0; программа резервного копирования HP Colorado Backup for Windows 95.

**Инструментальное ПО** (Software tools) - программное обеспечение, используемое в ходе разработки, корректировки или развития других программ: редакторы, компиляторы, отладчики, вспомогательные системные программы, графические пакеты и др.

**1. Системы программирования** - это набор специализированных программных продуктов, которые являются инструментальными средствами разработчика. Программные продукты данного класса поддерживают все этапы процесса программирования, отладки и тестирования создаваемых программ.

Множество различных приложений на компьютере создаётся с помощью языков и систем программирования.

**Язык программирования** - это формализованный язык описания алгоритмов, используемых для решения различных задач на компьютере.

Популярные системы программирования — Turbo Basic, Quick Basic, Turbo Pascal, Turbo C.

Современные системы программирования обычно предоставляют пользователям мощные и удобные средства разработки программ.

В них входят:

- 1) Компилятор или интерпретатор;
  - 2) Интегрированная среда разработки;
  - 3) Средства создания и редактирования текстов программ;
  - 4) Библиотеки стандартных программ и функций;
  - 5) Отладочные программы, т.е. программы, помогающие находить и устранять ошибки в программе;
  - 6) Многооконный режим работы;
  - 7) Графические библиотеки; утилиты для работы с библиотеками
  - 8) Встроенный ассемблер;
  - 9) Встроенная справочная служба;
- и другие специфические особенности.

Любая система программирования может работать только в соответствующей ей ОС, под которую она создана, однако при этом она может позволять разрабатывать программное обеспечение и под другие ОС.

Для того, чтобы компьютер мог понять программу, написанную на каком-то языке программирования, необходим переводчик (транслятор) такой программы в машинные коды.

**Трансляторы языка программирования** – это программа, предназначенная для преобразования программ, написанных на языках программирования, в машинный код.

Трансляторы делятся на два класса: **компиляторы** и **интерпретаторы**.

**Компилятор** преобразует (транслирует) всю программу в модуль на машинном языке, после этого программа записывается в память компьютера и лишь потом исполняется.

**Интерпретатор** - это транслятор, производящий покомандную обработку и выполнение исходной программы. Интерпретатор в отличие от транслятора не выдает результирующую программу или код.

**Ассемблеры** переводят программу, записанную на языке ассемблера (автокода), в программу на машинном языке.

**Прикладные программы** (Application software) - это комплекс программ, предназначенных для решения определенного класса задач. Основное назначение – дать пользователю средство обработки информации, которое не требует знаний языков программирования.

В состав **Прикладного ПО** входят:

- пакеты прикладных программ различного назначения
- рабочие программы пользователя и ИС в целом.

ППП является мощным инструментом автоматизации решаемых пользователем задач, практически полностью освобождая его от необходимости знать, как выполняет компьютер те или иные функции и процедуры по обработке информации.

Различают следующие типы ППП:

1) **Общего назначения (универсальные)** – это универсальные программные продукты, предназначенные для автоматизации разработки и эксплуатации функциональных задач пользователя и ИС в целом.

К этому классу относятся:

- редакторы текстовые и графические (MSWord, Paintbrush, PhotoShop)
- электронные таблицы (MS Excel, Lotus, Quattro Pro)
- системы управления базами данных (СУБД) (MSAccess, MSFoxPro)
- интегрированные пакеты, объединяющие в себе функционально различные программные компоненты ППП общего назначения (MSOffice)
- CASE-технологии (совокупность средств автоматизации разработки информационной системы, включающей в себя методологию анализа предметной области, проектирования, программирования и эксплуатации ИС, применяются при создании сложных ИС)
- экспертные системы и системы искусственного интеллекта (ЭС - это системы обработки знаний в узкоспециализированной области подготовки решений пользователей на уровне профессиональных экспертов)

2) **Мультимедиа** - одновременное использование в едином объекте различных форм представления информации и её обработки. Например, в одном объекте может содержаться текстовая, аудиальная, графическая и видео информация, а также, возможно, способ интерактивного взаимодействия с ней.

3) **Методо-ориентированные ППП** - включает программные продукты, обеспечивающие, независимо от предметной области и функции информационных систем, математические, статические и другие методы

решения задач. Наиболее распространены методы математического программирования, решение дифференциальных уравнений, имитационного моделирования, исследования операций.

4) **Проблемно-ориентированные ППП** - это программные продукты, предназначенные для решения какой-либо задачи в конкретной функциональной области.

Выделяются группы ППП для комплексной автоматизации функций управления:

- в промышленной сфере (Oracle, PRISM, Галактика (Россия))
- в непромышленной сфере (банковские, финансовые, правовые ППП)
- отдельных предметных областей (ППП бухгалтерского учета (1С.Бухгалтерия, Офис), ППП финансового менеджмента (ЭДИП, Инвестор), ППП правовых справочных систем (Консультант плюс, Гарант).

## 4.2. Операционные системы семейства Windows

Операционная система Windows компании Microsoft, вне сомнения, стала вехой в развитии не только информационной индустрии, но и всего человечества. Во многом именно благодаря Windows на столах сотен миллионов людей по всему миру установлены персональные компьютеры и ноутбуки. Благодаря Windows работа с компьютером стала доступна абсолютно всем, от детей дошкольного возраста до почтенных пенсионеров. Сложнейшие компьютерные системы, когда-то использовавшиеся только инженерами и учеными, в наше время применяются для самых разнообразных задач, работы и развлечения, учебы и познания мира.

Уже долгие годы Windows занимает в мире подавляющую долю на рынке операционных систем. На февраль 2009 года доля Windows составляла более 88,41%. Ее ближайшему конкуренту, операционной системе Mac OS, устанавливаемой на компьютерах компании Apple, досталось 9,61%, а операционной системе Linux - жалкие 0,88%. Иными словами, говоря "домашний" или "рабочий" компьютер, мы явно подразумеваем компьютер, на котором установлена та или иная версия Windows.

Операционная система семейства Windows представляет собой:

1. **Высокопроизводительная** - работает быстрее других.  
**Многозадачная**- может работать с несколькими программами одновременно (записать их в ОЗУ).

3. **Многопоточная** - позволяет при задержке в выполнении одного потока команд, решающих частную задачу внутри общей задачи, работать со следующим потоком.

4. **32-64 разрядная система** - ядро содержит 32 или 64 разрядный код, только некоторые модули имеют 16-разрядный код для совместимости с режимом MS-DOS.

5. **Объектно-ориентированная** - пользователь работает с документами, а программа (приложение) рассматривается как инструмент для работы с документом.

6. **Графический пользовательский интерфейс** - набор графических средств для взаимодействия пользователя и компьютера.

### 4.3. Основные элементы графического интерфейса Windows

Графическая оболочка ОС Windows обеспечивает взаимодействие пользователя с компьютером в форме диалога с использованием ввода и вывода на экран дисплея графической информации, управления программами с помощью пиктограмм, меню, окон, панелей (управления, задач, инструментов) и других элементов управления.

Основными элементами графического интерфейса **Windows** являются:

#### 1. Рабочий стол с пиктограммами (значками) и ярлыками.

В ОС Windows экран компьютера называется **Рабочим столом**.

**Рабочий стол** – это графическая среда, на которой отображаются объекты Windows (программы, файлы, документы) в виде **значков** (пиктограмм) и ярлыков, и элементы управления. На рабочем столе можно расположить произвольное количество пиктограмм (ярлыков). Вид Рабочего стола и набор элементов зависит от настройки компьютера и могут быть изменены пользователем.

Примеры значков рабочего стола: Корзина, сеть, компьютер.

**Значки** - это графические изображения, соответствующие одному объекту, которые позволяют быстро распознать и запустить нужную программу. Стандартный (установленный системой) рисунок значка отображает тип объекта (папка, текстовый документ, графический файл и т.д.). Подпись к значку и рисунок значка могут быть изменены пользователем.

Некоторые значки автоматически попадают на рабочий стол в тот момент, когда вы устанавливаете те или иные программы. Если в левом нижнем углу значка находится стрелка, значит, это **ярлык**.

**Ярлык** - это очень маленький по объему файл (менее 1 Кбайта), в котором в качестве информации содержатся **ссылки на объект** (документ, программа и т.д.), содержащую точный адрес объекта. Удаление или копирование ярлыка не влияет на сам объект, на которые он указывает. Отличить ярлыки от других объектов можно по маленькой черной стрелочке в левом нижнем углу значка.

Большинство программ автоматически создают ярлыки при установке. Вы можете создавать собственные **ярлыки** для программ, отдельных файлов, папок, сетевых ссылок или сайтов. Windows присваивает изображение каждому ярлыку в соответствии с типом элемента, на который он указывает.

Любая расположенная на рабочем столе пиктограмма (или ярлык) может быть удалена с него. Исключение из этого правила составляют лишь

пиктограммы, созданные операционной системой, такие, как **Компьютер**, **Сеть**, **Корзина**.

**Компьютер** – позволяет просмотреть содержимое компьютера, двойным щелчком раскрывая папку выбранного объекта.

**Сеть** – используется для просмотра имеющихся сетевых ресурсов, если компьютер подключен к локальной сети.

**Корзина** – предназначена для временного хранения удаленных файлов, позволяет восстановить ошибочно удаленные файлы. Для полного удаления файлов нужно очистить Корзину.

**2. Панель задач**, представляет собой длинную горизонтальную полосу в нижней части экрана. В отличие от рабочего стола, который может быть заслонен открытыми окнами, панель задач видна почти всегда. Она состоит из трех основных частей.

- кнопка Пуск
- средняя часть, которая отображает открытые программы, файлы, папки, позволяя быстро переключаться между ними;
- Область уведомлений (Панель индикации), на которой находятся часы и индикаторы состояния системы и значки (миниатюрные изображения), показывающие состояние некоторых программ и параметров компьютера.

**3. Главное меню** (меню Пуск), которое обеспечивает доступ практически ко всем ресурсам системы и содержит команды

**4. Контекстное меню** (отображается при щелчке правой кнопкой мыши по выбранному объекту) - меню, содержимое которого зависит от контекста, в котором вызвано меню, т.е. от вида и состояния объекта, с которым оно связано.

### **5. Окна Windows.**

Важнейшим элементом графического интерфейса Windows являются окна. Открываемые программы, файлы или папки появляются на экране в полях или рамках, называемых окнами (именно от них операционная система Windows - Окна - получила свое название).

**Окно** – это ограниченная рамкой область экрана, содержащая стандартные элементы, в которой представляется информация определенного свойства.

Окна можно открывать (разворачивать), закрывать, сворачивать, перемещать, упорядочивать, менять размеры. Открытое окно может занимать целый экран или его часть.

**Закреть окно** — значит полностью убрать его с экрана. Закрытие программного окна означает удаление программы из оперативной памяти.

**Свернуть окно** отображается в виде кнопки на Панели задач. Программа, окно которой свернуто, остается в оперативной памяти и с ней в любой момент можно возобновить работу. Чтобы вновь раскрыть свернутое окно нужно щелкнуть на кнопке на Панели задач.

В Windows поддерживаются окна 4-х типов:

1. **Окна дисков и папок** - отображается содержимое дисков и папок. Любую папку Windows можно открыть в своем окне. С помощью окон папок можно просмотреть всю файловую структуру дисков. В строке заголовка указывается имя папки, ниже располагаются меню, панель инструментов.

2. **Программные окна** (окна приложений)

Это окна, в которых работают загруженные в оперативную память Windows-программы (а возможно и DOS-программы). В строке заголовка – имя программы, ниже – строка меню, панель инструментов (может быть не одна), линейка. Внутри этих окон открываются окна документов.

3. **Окна документов** (вторичные окна) располагаются только в пределах своего программного окна, не имеют строки меню и могут быть открыты только в окне своего приложения.

4. **Окна запросов** (диалоговые) - в них содержится запрос какой-либо информации от пользователя или подтверждении его действий.

#### 4.4. Операции с файловой системой Windows

Основным носителем информации в операционной системе является **файл**, представляющий собой физическую область на диске, имеющую некоторые атрибуты (свойства) и помеченную именем. Содержанием файла может быть программа, текстовый документ, рисунок, видеоклип, звукозапись и др.

**Файл** – это именованная совокупность любых данных, размещённая на внешнем запоминающем устройстве и хранимая, обрабатываемая и перемещаемая как единое целое.

Совокупность правил и программ, по которым выполняются операции с файлами, называется файловой системой. Основные из них: **FAT** (File Allocation Table) – используется в операционных системах для 16-и и 32-х разрядных процессоров, **NTFS** (New Technology File System) – используется, начиная с ОС Windows NT для 32-х и 64-х разрядных процессоров). В них пользователю предоставляются следующие возможности:

- создание папок;
- копирование, перемещение, переименование и удаление файлов и папок;
- навигация по файловой структуре
- запуск программ и открытие документов;
- создание ярлыков;
- стандарты для обозначения пути к файлу

Основными атрибутами файла являются:

1. Собственное имя – строка букв и цифр. Предельная длина имени файла составляет 255 символов, включая пробелы. Имена не должны содержать следующих символов: / \ : \* ? “ < > | ; 2. Тип (расширение) – указывает тип файла. Расширение записывается через точку после имени файла (например: .txt – текстовый документ; .bmp – графический файл и т.д.).

3. размер – размер файла в байтах.

4. Дата создания или изменения – содержит дату и время создания (последнего изменения) файла.

Кроме перечисленных выше атрибутов каждому файлу сопоставляются атрибуты, определяющие допустимые действия с ним. Например:

– только для чтения (исправления, сделанные во время просмотра, не будут сохраняться после закрытия файла;

– скрытый (не будет высвечиваться в каталоге), но открыть его можно, введя имя вручную;

– архивный (для автоматического обновления изменённых версий в архивах);

– системный.

В операционной системе Windows имя файла уникально. На диске файл хранится в одном или нескольких фрагментах, называемых **кластерами**.

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

Файлы объединяются в каталоги (директории), которые в системе Windows называются **папки**. **Папкой (каталогом)** называется место на диске, в котором хранится информация о файлах, их именах, размерах, атрибутах и предназначена для группировки данных.

Папки делятся на два основных типа: файловые и объектные (системные). **Файловые папки** являются изображением каталогов на диске, содержат другие папки и файлы (Документы, Корзина). **Объектные папки** являются изображением логических и физических устройств компьютера (Рабочий стол, Мой компьютер, Сетевое окружение).

Папки имеют иерархическую структуру – **дерево папок**.

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

Просмотреть файловую структуру можно при помощи программы Проводник.

**Проводник** – это служебная программа Windows, предназначенная для навигации по файловой структуре компьютера и её обслуживания, цель которой – обеспечение доступа к нужной папке и её содержимому. Служит для работы с файлами, папками и устройствами компьютера, в которой графически с помощью пиктограмм и линий обозначается взаимное расположение объектов.

#### 4.5. Вопросы для самопроверки по теме 4

*Задание №1.* Укажите, какой из перечисленных терминов обозначает программы, обеспечивающие взаимодействие ОС с периферийными устройствами:

1. контроллер;
2. транслятор;
3. драйвер;
4. компилятор
- 5.

*Задание № 2.* Укажите, как называется именованная область внешней памяти произвольной длины с определённым количеством информации.

*Задание № 3.* Укажите, что не входит в основные функции операционной системы:

1. обеспечение диалога с пользователем;
2. разработка программ для ЭВМ;
3. управление ресурсами компьютера;
4. организация файловой структуры.

*Задание № 4.* Укажите, что относится к основным компонентам системного программного обеспечения:

1. обрабатывающие программы и система автоматизации программирования,
2. операционная система,
3. монитор и супервизор,
4. пакеты прикладных программ.

*Задание №5.* Укажите, какая группа файлов будет выделена по маске <\*. \*|\*.bak>.

*Задание № 6.* Укажите, к какому классу языков программирования относится Ассемблер

*Задание № 7.* Укажите основные особенности трансляции в режиме интерпретации.

*Задание № 8.* По каким критериям нельзя искать файл средствами операционной системы Windows?

1. дата создания;
2. размер файла;
3. тип файла;
4. фамилия автора файла.

## 5. ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

### 5.1. Текстовые редакторы

**Текстовый редактор** – это прикладная программа, предназначенная для создания и изменения текстовых документов, а также просмотра их содержимого на экране, вывода на печать, поиска фрагментов текста и т.п.

**Текстовые процессоры** – это программы, имеющие средства создания, обработки и хранения документов различной степени сложности.

Если в качестве классифицирующего признака текстовых редакторов взять сложность структуры документа, то их можно разделить на следующие типы:

1. **Простые редакторы текстов** предназначены для создания и редактирования простых текстовых документов, содержащих неформатированный текст. Иногда их называют редакторы кода, так как основное их предназначение – написание исходных кодов компьютерных программ. К ним можно отнести:

- **Блокнот** – это несложный текстовый редактор, используемый для создания простых документов или веб-страниц, входящий в состав стандартных приложений семейства операционных систем MSWindows;

- **Notepad** – свободный текстовый редактор с открытым исходным кодом для Microsoft Windows, построен на принципах быть маленьким, быстрым и полезным;

- **EditPlus** – текстовый редактор для MSWindows, предназначенный для программирования и веб-разработки, имеет возможность перевода текста из кодировки DOS в кодировку Windows и обратно;

- **Pspad** – текстовый редактор и редактор исходных текстов программ для операционной системы Windows с простым управлением и мощными возможностями редактора кода, необходимый для работы одновременно с разными языками программирования и т.д.

2. **Редакторы документов** (их ещё называют текстовыми процессорами) предназначены для создания и редактирования сложных по структуре документов, состоящих из страниц, абзацев, вложенных документов. В них могут быть внедрены таблицы, формулы, графические объекты, зачастую созданные в других приложениях. К ним относятся:

- **WordPad** – процессор, предназначенный для создания и редактирования текстовых документов, содержащих форматирование или рисунки, входящий в состав стандартных приложений операционных систем MS Windows;

- **MSWord** – текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов;

- **WordPerfect** – программа для электронной обработки текстов, доступная для целого ряда компьютеров и операционных систем. Является альтернативой пакету MicrosoftOffice. Пакет включает в себя программы для обработки текстов, электронных таблиц, создания презентаций, баз данных и графики.

3. **Редакторы научных текстов** - обеспечивают подготовку и редактирование научных текстов, содержащих большое количество математических формул, графиков, специальных символов и т. д. (**TEX** и **MathOr.**).

4. **Издательские системы** - используются для подготовки больших сложных документов (книги, альбомы, журналы, газеты, буклеты). (**Corel Ventura Publisher, Adobe PageMaker, QuarkXPress**).

Пакет **Word** предназначен для создания и редактирования разного рода документов, включая отчеты, письма, наклейки и пр. Помимо работы с текстом, пакет имеет довольно простые средства работы с рисунками и таблицами. Кроме того, этот пакет позволяет использовать данные других приложений, таких, как *Excel, Access, Internet Explorer, Outlook* и др.

В настоящее время существует несколько версий этого пакета. Наиболее известным являются Word 6.0, Word 7.0 (пакет Microsoft Office 97), Word 8.0 (пакет Microsoft Office 2000), Word XP (пакет Microsoft Office XP), Word 2003, Word 2007, Word 2010 и Word 2013. Существовали также различные пакеты для работы в системе ДОС, но работа в них практически аналогична. Здесь мы будем рассматривать Word версии 2010 для Windows, в дальнейшем вместо Word 2010 мы будем писать просто Word. Word 2000, Word XP и Word 2003 довольно близки друг к другу. В этом можно убедиться, посмотрев различия между Word 2000 и Word XP. Следующая версия Word 2007 резко отличается от более ранних версий, так как вместо верхней строки меню появились вкладки с режимами. Режимы для работы несколько перепутались, и для перехода с версии 2003 на 2007 требуется время. Версии 2010 и 2013 довольно близки версии 2007, поэтому и описание данных пакетов близко описанию версии 2007.

## **5.2. Текстовый редактор WORD, назначение и возможности. Основные средства текстового редактора WORD 2010**

### **РАЗМЕЩЕНИЕ РЕДАКТОРА**

Текстовый редактор WORD – это комплекс программ объемом около 2,92 Кбайт, на диске 4 КБ, расположенных в отдельной папке либо в общей папке пакета MSOffice. Главный файл – Winword.exe.

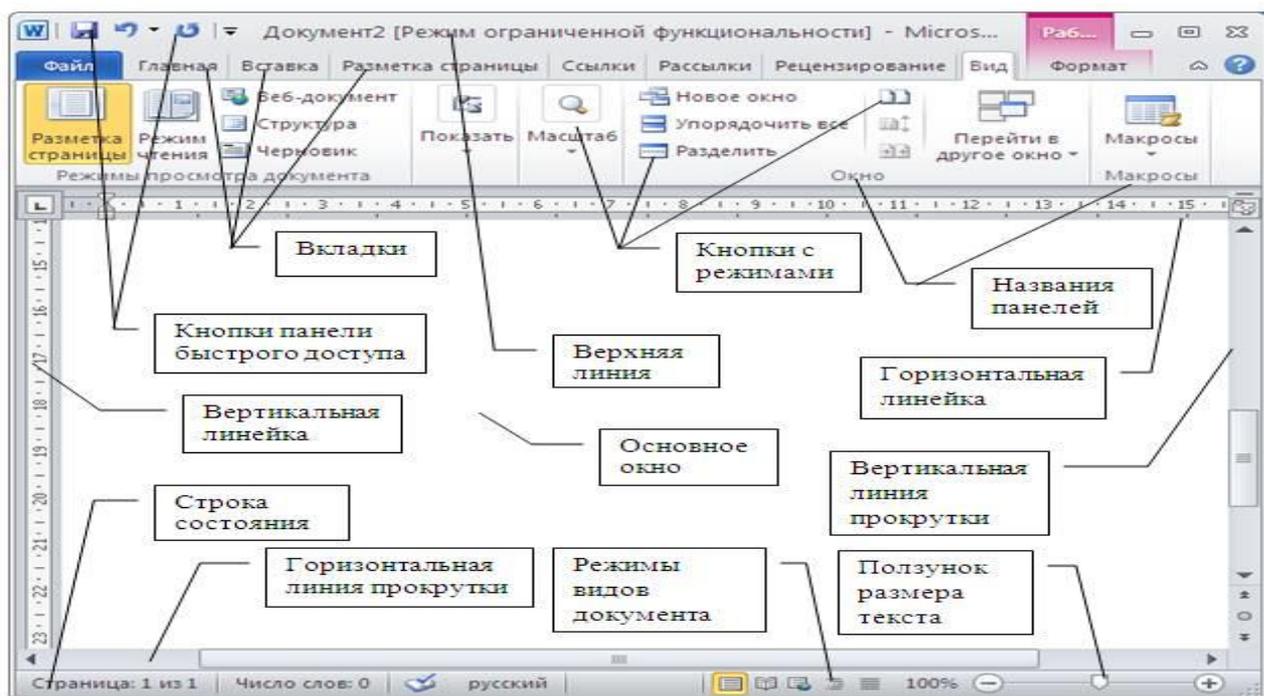
### **ЗАПУСК WORD**

Запустить MSWord, можно одним из перечисленных ниже способов:

1. Выбрать соответствующую команду в Главном меню (**Пуск** → **Все программы** → **MicrosoftOffice** → **Word 2010**).

2. При помощи ярлыка *MicrosoftWord*, расположенного на **Рабочем столе**.
3. С помощью кнопки *MicrosoftWord*, расположенной на панели быстрого запуска (кнопка Пуск).
4. Открыв любой документ, созданный этой программой.

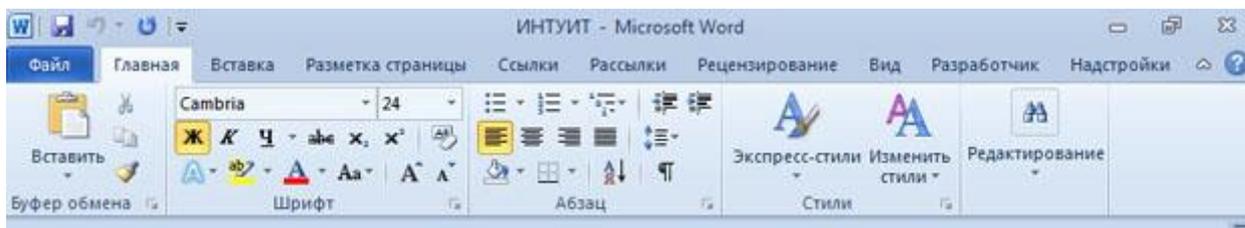
При работе с документом рекомендуется чаще использовать режим сохранения (Файл→Сохранить), особенно когда вы имеете дело с сложным документом, содержащим много графики, с тем чтобы, если вдруг выключится электропитание или произойдет сбой программы, были сохранены последние изменения.



Окно состоит из

- **строки заголовка**, на которой написано имя программы (Документ2), выполняемой в данный момент. По правому краю имеются кнопки для открытия окна. В левой части находится кнопка для работы с окном;
- **панели быстрого запуска** содержит наиболее часто используемые кнопки, которые можно изменять;
- **Лента**, на которой находятся: вкладки меню и панели инструментов. С помощью ленты можно быстро находить необходимые команды (элементы управления: кнопки, раскрывающиеся списки, счетчики, флажки и т.п.). Команды упорядочены в логические группы, собранные на вкладках;
- **рабочая область** – текстом документа, где происходит работа;
- **вертикальной линии прокрутки** или **полосы прокрутки** (справа от основного окна), позволяющей просматривать те части документа, которые не поместились на экране сверху и снизу;
- **масштабных линеек** с указанием размера документа по вертикали и по горизонтали;

- **горизонтальной линии или полосы прокрутки** (внизу от основного окна), позволяющей просматривать те части документа, которые не поместились на экране слева и справа;
  - **строки состояния**, где отображается текущее состояние редактора;
  - **ползунок**, при помощи которого можно изменять размер текста документа.
- Кроме того, в основном поле находятся курсор в виде вертикальной черточкой для ввода текста, курсор мыши в виде стрелочки.



Главный элемент пользовательского интерфейса Microsoft Word 2010 представляет собой ленту, которая идет вдоль верхней части окна каждого приложения, вместо традиционных меню и панелей инструментов. С помощью ленты можно быстро находить необходимые команды (элементы управления: кнопки, раскрывающиеся списки, счетчики, флажки и т.п.). Команды упорядочены в логические группы, собранные на вкладках.

Заменить ленту панелями инструментов или меню предыдущих версий приложения Microsoft Word нельзя.

Удалить ленту также нельзя. Однако, чтобы увеличить рабочую область, ленту можно скрыть (свернуть).

Нажмите кнопку **^**, расположенную в правой части линии названий вкладок, и выбрать пункт **Свернуть ленту**. Лента будет скрыта, названия вкладок останутся. Ленту можно свернуть и иначе.

В контекстном меню любого места ленты выберите команду **Свернуть ленту**

Чтобы свернуть или восстановить ленту, можно также нажать комбинацию клавиш **Ctrl + F1**.

По умолчанию в окне отображается восемь постоянных вкладок: **Файл**, **Главная**, **Вставка**, **Разметка страницы**, **Ссылки**, **Рассылки**, **Рецензирование**, **Вид**. Для перехода к нужной вкладке достаточно щелкнуть по ее названию (имени).

Каждая вкладка связана с видом выполняемого действия. Например, вкладка **Главная**, которая открывается по умолчанию после запуска, содержит элементы, которые могут понадобиться на начальном этапе работы, когда необходимо набрать, отредактировать и отформатировать текст. Вкладка **Разметка страницы** предназначена для установки параметров страниц документов. Вкладка **Вставка** предназначена для вставки в документы различных объектов.

Для удобства работы с документом в **MS Word** можно осуществлять в **различных режимах** просмотра в зависимости от выполняемых задач:

1. **Режимы документов**, используются при работе над документами **Вид-Режимы просмотра документа** или :

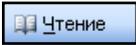
- **Режим разметки страницы** позволяет получить адекватное представление о странице документа. Текст, рисунки и другие элементы отображаются так же, как после вывода страницы на печать. Этот режим используется для редактирования колонтитулов, настройки полей, работы со столбцами и графическими объектами. Для перехода в этот режим используется команда меню **Вид→Разметка страницы**.

- **Режим структуры документа** предоставляет удобные средства для работы со структурой документа. В режиме структуры можно свернуть документ, оставив основные заголовки, или развернуть его, отобразив все заголовки и основной текст. Границы страниц, колонтитулы, рисунки и фон в этом режиме не отображаются. Для перехода в этот режим используется команда меню **Вид→Структура**.

- **Режим Web-документа** используется для создания веб-страниц или документов, предназначенных для просмотра на экране. Предлагается для применения в Internet. В этом режиме отображается фон, перенос текста выполняется по размерам окна, а рисунки занимают те же позиции, что и в окне веб-обозревателя. Для перехода в этот режим используется команда меню **Вид→Веб-документ**.

- Все режимы могут сочетаться для удобства пользователя, а также применяться независимо друг от друга.

2. **Режимы просмотра**, используются для чтения документа и перемещения в нём:

- **Режим чтения** используется для улучшения условий чтения документов на экране с наименьшим напряжением для глаз и оптимизированными для чтения инструментами. В этом режиме не отображаются на экране лишние панели инструментов и линейки. В режиме чтения документ отображается не в том виде, в котором он будет выведен на печать. Для перехода в этот режим используются команда меню **Вид→Режим чтения** и кнопка  в группе **режимы просмотра документа**.

- **Режим предварительного просмотра**, позволяет просматривать сразу несколько страниц в том виде, как они будут выведены на печать. В этом режиме можно просматривать разрывы страниц и подложку, а также изменять содержимое или форматирование документа до его вывода на печать. Для перехода в этот режим используются команда меню **Файл→Печать**

- Word позволяет установить **масштаб изображения** при помощи команды меню **Вид-Масштаб** или масштабного поля в строке состояния

**Создание документа в Word 2010** производится простым открытием нового документа и вводом текста в него. **Файл/Создать/Новый документ** в результате чего откроется пустой документ базового типа. При

необходимости вы можете создать документ особого типа, тогда можно воспользоваться одним из специально разработанных шаблонов.

Шаблоны — это превосходный способ сэкономить время при создании единообразных документов в Office 2010. Это особенно актуально для типов документов, которые часто используются, например еженедельных презентаций, заявлений и отчетов о расходах. В таких случаях следует использовать файл, уже имеющий необходимое оформление с заполнителями, которые можно менять в соответствии с текущими потребностями. Такой файл и есть шаблон, в котором сложная работа уже выполнена, оградив вас от необходимости начинать все с чистого листа.

**Шаблон документа** — это файл с расширением dot, в котором содержатся набор стилей, а также настройки меню, панелей инструментов и сочетаний клавиш, элементы автотекста и макросы. Шаблон может также включать текст, предназначенный для вставки в создаваемые на его основе документы. **Файл/Создать/** выберите один из имеющихся шаблонов, также дважды щелкнув по нему кнопкой мыши. Помимо простого использования шаблонов вы также можете вносить в них различные изменения и сохранять для последующего использования.

С понятием шаблона связано понятие стиль:

**Стиль** — это набор форматизирующих команд, сохраняемый под своим именем для многократного использования, упрощает задачу создания и оформления документа, облегчает изменение вида документа, экономит время форматирования.

Word обладает большим числом стандартных, или встроенных, стилей. Список доступных стилей зависит от шаблона и перечислены в окне списка на вкладке **Главная/Стили/Изменить Стили**.

#### ОТКРЫТИЕ ДОКУМЕНТА

Существует несколько способов открытия уже существующего документа Word:

1. Двойной щелчок по **значку** документа на Рабочем столе или в папке.
2. Команда **Документы** в Главном меню.
3. Непосредственно из программы Word: команда **Файл→Открыть** или соответствующая кнопка на **Стандартной** панели инструментов (в виде открытой папки).
4. Команда **Файл**, в нижней части открывшегося меню - список нескольких последних (3-4) открывавшихся документов.

#### СОХРАНЕНИЕ ДОКУМЕНТА

Пока документ не сохранен, все изменения находятся только в оперативной памяти компьютера. Сохраненный документ хранится в файле на диске. Для сохранения документа можно использовать команды

**Файл→Сохранить, Файл→Сохранить как...** Здесь же можно создать папку с помощью соответствующей кнопки.

Можно также установить **режим автосохранения**, при котором активный документ автоматически сохраняется через определенные промежутки времени (команда **Файл→Параметры**).

В Word можно работать с несколькими документами одновременно. Каждый создаваемый или открываемый документ размещается на экране в своём собственном окне. Окно, в котором работает пользователь, является **активным**. Оно располагается поверх других окон и имеет выделенный заголовок. Для переключения между окнами используется команда меню **Окно** или кнопка, соответствующая документу, на панели

При задании имени файла необходимо следовать определенным правилам:

1. Имя файла **MS Word** может иметь длину до 255 символов, включая пробелы.
2. Можно использовать прописные и строчные буквы (A — Z, a — z, А — Я, а — я), цифры (0 — 9), а также специальные символы.
3. В имени **недопустимы**: звездочки (\*), двоеточия (:), точка с запятой (;), знак вопроса (?), символы наклонной черты ( / или \ ), разделители ( | ), знаки больше чем ( > ) и меньше чем ( < ), кавычки (").
4. Из числа специальных можно использовать следующие символы: !, @, #, \$, % ,&, (, ), \_, -, { и }. Точки допустимы только в качестве разделителя между именем и расширением файла.

#### ЗАКРЫТИЕ ДОКУМЕНТА

1. Команда **Файл→Заккрыть**.
2. Кнопка **Заккрыть** (X) в строке заголовка активного документа.
3. Щелчок по системному значку документа в строке заголовка, команда **Заккрыть**.
4. Сочетание клавиш **Alt+F4** (закрывает любое открытое окно).

При необходимости **Word** даст возможность сохранить последние изменения в диалоговом окне, соответствующим вопросом.

#### Печать документа

С помощью команды **Файл→Печать** можно выполнить печать отдельных фрагментов документа и нескольких его копий. Быстрая печать на панели быстрого доступа. На панели быстрого доступа – **Быстрая печать**. Напечатает 1 экз.

**Основными этапами подготовки текстовых документов** являются:

1. Набор текста.
2. Редактирование текста.
3. Форматирование текста.
4. Печать текста.

## ВВОД ТЕКСТА

**Ввод текста** заключается в последовательном выводе на экран каждого символа текста. Текст вводится с помощью алфавитно-цифровых клавиш.

Для ввода прописных букв используется клавиша *Shift*.

Для фиксации режима ввода заглавных букв служит клавиша *CapsLock*.

Для переключения с русского на латинский алфавит необходимо использовать комбинации клавиш *Ctrl+Shift* или *Alt+Shift*.

При вводе текста необходимо соблюдать следующие правила:

1) Во всех современных текстовых процессорах переход на новую строку и новую страницу в процессе набора текста происходит автоматически, как только заполняется текущая строка или страница, не требуя ввода специального символа;

2) Не следует начинать абзац с пробелов;

3) Между словами ставиться только один пробел;

4) Клавиша *Enter* используется только для перехода к следующему абзацу, позволяющая перейти на новую строку;

5) Пробел обязателен после знака препинания, перед знаком препинания пробел не ставится;

6) Перед знаками « “ ( [ { ставится пробел, следующее за этими знаками слово пишется без пробела. Знаки » ” ) ] } пишутся слитно со словом, за которым следуют, после этих знаков ставится пробел;

7) Знак дефиса в сложных словах (-) пишется слитно с предшествующей и последующей частями слова;

8) Для образования знака тире (–) до и после знака ставиться пробел.

## РЕДАКТИРОВАНИЕ ТЕКСТА

**Редактирование** (от лат. *redactus* приведённый в порядок) – это внесение каких-либо изменений в содержимое документа. Обеспечивает добавление, удаление, перемещение символов, слов, строк и фрагментов текста, выявление и устранение ошибок в тексте, проверку правописания. Редактирование документа осуществляется как в процессе ввода текста, так и после его ввода.

Существуют два режима ввода и редактирования текста:

1. **Режим вставки** – вновь вводимые символы отодвигают вправо текст документа, стоящий за курсором, не удаляя его.

2. **Режим замены** – вновь вводимые символы замещают символы текста документа.

Режимом вставки пользуются, когда текст вводится впервые, а также при редактировании текста, когда добавляется в текст символ, слово или предложение. Если же надо заменить один символ другим, используют режим замены.

Переключать режимы с помощью кнопки **Insert**.

### **Проверка правописания**

WORD позволяет реализовать два режима проверки правописания -

автоматический и командный. Для работы в автоматическом режиме нужно установить соответствующие флажки с помощью команды **Рецензирование/Правописание**. В этом режиме при вводе текста слова, содержащие **орфографические** ошибки, подчеркивается **красной** волнистой линией, так же это может означать, что данного слова нет во встроенном словаре программы. **Грамматическая** или **стилистическая** ошибка помечается **зеленой** волнистой линией (пропущенная в тексте запятая, чрезмерная длина предложения и т.д.).

Характер ошибки можно узнать с помощью контекстного меню отмеченного слова или фрагмента. Это меню также содержит команды редактирования: пропустить (т.е. не исправлять); добавить (добавить данное слово во встроенный словарь); заменить (на один из предлагаемых вариантов).

## ФОРМАТИРОВАНИЕ ДОКУМЕНТА

**Форматирование документа** – это изменение его внешнего вида текста, при котором не изменяется его содержание.

MS WORD обеспечивает форматирование документов на пяти различных уровнях:

1. На уровне символов.
2. На уровне абзацев.
3. На уровне страниц.
4. На уровне разделов.
5. На уровне документа.

**!!!Объект форматирования должен быть предварительно выделен!!!** При форматировании отдельного слова, абзаца или раздела можно просто установить курсор в любом месте слова, абзаца или раздела.

**Форматирование на уровне символов** (понятие «символ» включает в себя не только отдельный символ, но слово, фразу, а также фрагмент текста, не являющийся абзацем) осуществляется при помощи команды меню **Главная→Шрифт** или кнопками из блока **Шрифт**.

Основные приёмы форматирования символов включают выбор и изменение:

1. Гарнитуры шрифта (стандартными для использования являются TimesNewRoman и Arial) **Главная/блок Шрифт**;
2. Начертания (обычный, курсив, полужирный и полужирный курсив) **Главная/блок Шрифт**;
3. Размеры (кегля), цвета шрифта и наличием типа подчеркивания (стандартными для использования являются 12 или 14 пунктов и черный цвет) **Главная/блок Шрифт**;
4. Межбуквенного интервала (обычный, уплотненный и разреженный) вкладка **Шрифт/Дополнительно/интервал**;
5. Анимация на блоке **Шрифт/(А)раскрывающийся список Анимация**

Для изменения регистра шрифта (как в предложениях, все строчные, все прописные, начинать с прописных) используют команду на блоке **Шрифт/раскрывающийся список → Регистр(Aa)**.

Для обнаружения ошибок форматирования можно установить режим **Непечатаемые знаки** при помощи кнопки  на блоке Абзац вкладка Главная.

**Абзац** – это фрагмент текста, процесс ввода которого заканчивается нажатием на клавишу ввода **Enter** и заканчивается непечатаемым символом ¶. К абзацу относятся: абзац основного текста, заголовки, оглавление, списки (нумерованные и маркированные) и т.д.

**Форматирование на уровне абзацев** осуществляется:

При помощи команды **Вкладка Главная → Абзац** или с помощью контекстного меню:

1. Выбор и изменение **отступов и интервалов:**

- Выравнивание (по левому и по правому краю, по центру и по ширине);
- Отступы справа и слева;
- Отступ и выступ первой строки (красной строки);
- Отступы до и после абзаца;
- Междустрочный интервал (стандартным является одинарный).

2. Управление размещением абзацев на страницах **Абзац/(вкладка**

**Положение на странице):**

- Запрет висячих строк;
- Не разрывать абзац;
- Не отрывать от следующего;
- Начать с новой страницы;
- Запретить нумерацию строк;
- Запретить автоматический перенос слов.

При помощи команды меню **Главная → Абзац/Список** создание маркированных, нумерованных и многоуровневых списков абзацев.

При помощи команды меню **Главная → Абзац/Границы** включает оформление рамкой абзаца.

---

Если удалить символ конца текущего абзаца, то абзац принимает параметры форматирования абзаца, следующего за ним в документе. Новый абзац приобретает параметры форматирования предыдущего абзаца.

**Форматирование на уровне страниц** осуществляют:

1. При помощи команды меню **Файл/печать/ Настойка** (просмотр документа в том виде, как он будет напечатан) включает выбор и изменение:

1. Полей страницы (верхнее, нижнее, правое и левое) и ориентации страницы (книжная и альбомная);

2. При помощи команды **Разметка страницы → Фон страницы** включает выбор и изменение типа, цвета, ширины и рисунка рамки страницы.

3. При помощи команды меню **Вставка→Разрыв** осуществляет принудительный разрыв страницы.

**Раздел** – это часть документа, имеющая заданные параметры форматирования страницы (колонки, колонтитулы, сноски, поля, нумерация и ориентация страниц). Наличие разделов позволяет создавать сложные документы. Новый раздел создаётся при необходимости изменения нумерации строк, числа столбцов или колонтитулов.

**Форматирование на уровне разделов** осуществляют:

2. При помощи команды меню **Вставка→блк Страницы/Разрыв страницы** осуществляет принудительный разрыв нового раздела со следующей страницы, на текущей странице, с чётной и нечётной страницы.

3. При помощи команды **Вставка →Колонтитулы** можно создать колонтитулы.

1. **Колонтитулы** – это структурный элемент документа, которые находятся в верхней или нижней части каждой страницы, содержат некоторую информацию (текст, графику, номера страниц, дату, время и т.д.).

4. При помощи команды **Ссылки→Сноски** можно создать сноски.

**Сноска** – это примечание к какому-либо термину основного текста, которое можно поместить либо в нижней части текущей страницы, либо в конце документа.

5. При помощи команды меню **Разметка страницы/ Параметры страницы/Колонки** можно разбить весь текст, часть текста или раздел текста на колонки. Можно создавать различные варианты **колонок**: одинаковой ширины, разной ширины, одинаковой длины, разной длины, с разделителем, со встроенными рисунками.

Также для разделов можно установить разные поля для разных страниц. По умолчанию эти виды форматирования применяются ко всему документу.

**Форматирование на уровне документа**

1. При помощи команды **Вставка→ Колонтитулы/Номер страницы** осуществляют разбивку на страницы;

2. При помощи команды **Ссылки→ Оглавление** осуществляют создание оглавления. **Оглавление** представляет собой список заголовков документа и номеров страниц.

## ТАБЛИЦЫ В ТЕКСТОВЫХ ДОКУМЕНТАХ

**Таблица** представляет собой совокупность ячеек, расположенных в строках и столбцах, которые можно заполнять произвольным текстом или графикой. При вводе текста ячейка растягивается по вертикали, ширина не меняется. К содержимому ячейки можно применять все приёмы редактирования и форматирования текста.

### СОЗДАНИЕ ТАБЛИЦ

**Создать таблицу** в MSWORD можно 4 способами:

1-й способ: **автоматически** с помощью Мастера, при помощи команды **Вставка /Таблица→Вставить таблицу**. Открывается диалоговое

окно **Вставка таблицы**, в котором задают число строк и столбцов, а также автоподбор ширины столбцов (параметры для настройки размера таблицы).

2-й способ: **методом рисования** для создания таблиц сложной структуры, при помощи команды **Вставка /Таблица→Нарисовать таблицу**.

3-й способ: можно преобразовать текст в таблицу при помощи команды меню **Таблица→Преобразовать в таблицу**. Следует иметь в виду, что фрагмент текста должен быть предварительно отформатирован символами-разделителями, чтобы программа смогла различить столбцы таблицы.

4-й способ: выбрать одну из предварительно отформатированных таблиц с образцами данных. Чтобы вставить таблицу этим способом, поставьте курсор в нужное место документа, после чего во вкладке **«Вставка» /«Таблицы» / «Таблица» /«Экспресс-таблицы»**, выберите нужный шаблон из

раскрывающегося списка. Добавленный шаблон таблицы будет содержать примерные данные, которые служат для предварительного просмотра итогового вида таблицы. Замените их своими данными.

### **Редактирование таблиц**

Под редактированием таблиц, мы имеем в виду не редактирование их содержимого, а только редактирование их структуры. Редактирование содержимого осуществляется обычными средствами, рассмотренными ранее. Перед выполнением операций по изменению структуры таблицы, необходимо выделить соответствующие ячейки, строки или столбцы с помощью мыши или команд меню **Работа с таблицами / Макет→Выделить**. Все операции по редактированию таблиц **Работа с таблицами и выбираем пункты из вкладок Конструктор или макет**.

В редактирование структуры таблиц входят следующие операции:

1. Добавление заданного количества строк или столбцов;
2. Удаление выделенных ячеек, строк и столбцов;
3. Объединение или разбиение выделенных ячеек;
4. Изменение ширины столбцов и высоты строк.

Команды редактирования можно выполнить при помощи команд в контекстном меню.

**Изменить ширину столбца или высоту строки** можно также с помощью мыши методом перетаскивания, не выделяя их.

### **Форматирование таблиц**

При работе с таблицами следует различать форматирование таблиц и форматирование содержимого. В первом случае происходит управление размерами структурных элементов таблицы (ячеек, строк, столбцов и т. п.), а во втором – управление размещением содержимого ячеек.

**Форматирование таблиц** включает в себя:

1. Задание метода выравнивания таблицы относительно страницы документа при помощи панели инструментов **Форматирование: по левому и по правому краю, по центру и по ширине страницы**.

2. Назначение параметров текущей строки, столбца, ячейки или выделенных строк, столбцов, ячеек при помощи команды меню
3. Оформление внешних и внутренних рамок таблицы, заливки или штриховки
4. Разбиение таблицы для вставки текста при помощи команды меню;
5. Сортировка (упорядочивание) данных в таблице при помощи команды меню. Сортировать можно текст, числа или даты в порядке возрастания или убывания;
6. Нумерация строк или столбцов таблицы при помощи команды меню;
7. Вертикальное выравнивание в ячейке для выравнивания заголовков и содержимого таблицы при помощи команд контекстного меню **Выровнять в ячейке**.
8. Форматирование содержимого ячеек **АВТОМАТИЧЕСКАЯ** Выровнять высоту строк и ширину столбцов;
  1. Выровнять ширину столбцов по содержимому, по ширине окна;
  2. Подобрать фиксированную ширину столбцов.
  3. Направление текста.

#### ОБЪЕКТЫ В РЕДАКТОРЕ MS WORD

Главным достоинством современных процессоров является возможность создания сложных составных документов, включающих объекты разных типов, создаваемых с помощью других приложений (электронных таблиц, баз данных, графических редакторов и других программ). Объекты создаются в специально предназначенных приложениях и потом переносятся в документ как объект редактора. Перенос объектов в документ из других приложений может осуществляться двумя способами: **внедрением** или **связыванием**. Эти возможности обеспечиваются технологией OLE, поддерживаемой ОС Windows. Технология OLE (ObjectLinkingandEmbedding) позволяет устанавливать связь между приложениями, называемыми приложением-источником и приложением-приемником (в нашем случае текстовый редактор MSWord является приложением-приемником и в него передаются объекты из приложений-источников). Основное отличие между связанными и внедренными объектами заключается в месте их хранения (внедренный объект хранится в этом документе, при связывании в документ помещается ссылка на объект-источник и данные хранятся в исходном файле). Обновление **связанных данных** происходит только при изменении исходного файла. **Внедренный объект** становится частью конечного файла и, будучи однажды вставленным, больше не является частью исходного файла, при изменении исходного файла внедренный объект не изменится.

Для вставки объектов в редактор используются блоки из вкладки **Вставка (Иллюстрации, Текст, символы)**.

Приложение редактор формул *MSEquation* – позволяет вставлять математические формулы и выражения. Панель инструментов редактора формул содержит два ряда кнопок. Верхняя строка панели инструментов **Формула** содержит более 150 математических символов. Нижняя строка используется для выбора разнообразных шаблонов, предназначенных для построения дробей, интегралов, сумм и других сложных выражений. Заполнение этих полей может производиться как с клавиатуры, так и с помощью элементов управления верхней строки. Переходы между полями выполняются с помощью клавиш управления курсором.

Чтобы вставить формулу в документ, перейдите во вкладку «Вставка» /Объект/объект, открывается окно **Вставка объекта** и выбираете пункт *MSEquation 3.0*. Кроме того, можно выбрать готовую формулу. **Вставка/Симолы/Формула**

Приложение диаграмма *MSGraph* – позволяет вставлять диаграммы. Мастер диаграмм предлагает готовые таблицы и при замене предложенных параметров на свои, можно получить необходимую диаграмму.

#### ГРАФИЧЕСКИЕ ОБЪЕКТЫ В РЕДАКТОРЕ MSWORD

В документах MSWord можно использовать два тип графических объектов: **рисунки** и **изображения**.

**Рисунки** – это объекты векторной природы (линии, геометрические фигуры). Для работы с векторными рисунками служит вкладка Вставка/блоки Иллюстрации и Текст. Основным средством этой панели, предназначенным для создания простейших объектов, является раскрывающийся список *Автофигуры*. Размер создаваемых объектов можно редактировать путем перетаскивания маркеров выделенного объекта. Выделяется объект щелчком мыши. Свойства объекта (толщина линии, цвет, заливка, обтекание объекта) редактируются при помощи вставки **Формат**→*поле средства рисования* или с помощью контекстного меню.

Для создания текстовых элементов в поле рисунка используют команду **Вставка**→*Надпись*.

Несколько выделенных объектов можно сгруппировать в единую композицию командой контекстного меню *Группировка*.

**Изображения** – это растровые графические объекты, выполненные посторонними программными средствами или полученные из внешнего источника. Вставка изображений выполняется при помощи команды меню **Вставка**→*Рисунок*→Из файла, Картинки, Объект WordArt, Со сканера или камеры, Диаграмма и т.д. Редактирование изображений возможно в ограниченном варианте: изменение яркости и контрастности, обтекание текстом с помощью панели инструментов *Настройка изображения*; перемещение и изменение размеров с помощью мыши.

С помощью приложения *WordArt* в документе можно создать эффектные надписи.

Выделение графических объектов производится щелчком мыши. Выделенный объект можно перемещать, изменять его размеры, вызывать

контекстное меню. Двойной щелчок раскрывает создавшую объект программу и дает возможность редактирования объекта.

### 5.3. Электронные таблицы. Назначение и классификация

Программные средства для проектирования электронных таблиц – табличные процессоры - относятся к пакетам прикладных программ общего назначения. Они находят широкое применение во многих областях народного хозяйства, требующих обработки табличных данных. В настоящее время наиболее популярными электронными таблицами для персональных компьютеров являются табличные процессоры Excel (фирма Microsoft), Lotus 1-2-3 (фирма LotusDevelopment), QuattroPro (фирма WordPerfect – NovellApplicationsGroup). Между собой данные пакеты различаются набором выполняемых функций и удобством интерфейса.

Табличный процессор Excel занимает лидирующее положение на рынке электронных таблиц, его предпочитают почти 80% пользователей электронных таблиц.

**Электронные таблицы** (далее ЭТ) предназначены для хранения и обработки информации, представленной в табличной форме, позволяют создавать таблицы различной степени сложности и автоматизировать обработку табличных данных. К основным функциям табличных процессоров относятся:

1. Создание и редактирование электронных таблиц;
2. Оформление и печать электронных таблиц;
3. Создание многотабличных документов, объединенных формулами;
4. Построение диаграмм, их модификация и решение экономических задач графическими методами;
5. Работа с электронными таблицами как с базами данных;
6. Создание итоговых и сводных таблиц;
7. Решение оптимизационных задач;
8. Статистическая обработка данных;
9. Создание слайд-шоу;
10. Разработка макроканд, настройка среды под потребности пользователя.

### 5.4. Общие сведения о табличном процессоре MS Excel

Табличный процессор **MS Excel** – это инструмент для хранения, обработки и представления чисел и любой другой деловой информации. Табличный способ существенно упрощает выполнение финансовых, научных и других видов расчетов.

В **MS Excel** существует возможность оформления всех бухгалтерских и финансовых документов, а также возможность быстрого анализа данных и представление их в графическом виде. **Основу** программы составляют: вычислительно-калькуляционный модуль, модуль диаграмм, доступ к

внешним базам данных, модуль программирования для индивидуальных задач.

## ОСНОВНЫЕ ПОНЯТИЯ

В табличном процессоре Excel можно работать с 4 основными типами документов: рабочей книгой, электронной таблицей (рабочим листом), диаграммой, макротаблицей.

**Рабочая книга** в MSExcel - это файл, объединяющий несколько рабочих листов. Новые файлы создаются при помощи команды меню **Файл**→**Создать**. Новые файлы получают имена Книга1, Книга2, Книга3... и расширение **.xlsx**.

В одной рабочей книге может быть от 1 до 255 рабочих листов. **Рабочий лист** служит для организации и анализа данных. Можно вводить, исправлять и обрабатывать данные на нескольких листах одновременно. Каждый лист имеет уникальное имя и по умолчанию листы называются Лист1, Лист2, Лист3 и т.д., которое выводится на ярлычке в нижней части экрана. Различные действия с листом (переименование, добавление, перемещение, удаление и т.п.) можно произвести при помощи контекстного меню ярлычка рабочего листа или при помощи команд меню **Правка** и **Вставка**.

Каждый лист рабочей книги представляет собой электронную таблицу. **Электронная таблица** – это двумерный массив, состоящий из столбцов и строк. Столбцы идентифицированы буквами латинского алфавита (**A, B, C...**) и двойными буквами (**AA, AB, AC...**), строки идентифицированы арабскими цифрами (**1, 2, 3...**) (Рис.1). Табличный процессор Excel содержит 256 столбцов и 65536 строк.

	A	B	C	D	E	F	G	H	...
1									
2									
3									
4									
5									
...									

Рис. 1. Основные элементы электронной таблицы

Место пересечения столбца и строки называется **ячейкой**. Каждая ячейка имеет свой уникальный адрес, состоящий из имени столбца и номера строки, например **A5**, **G34** и т.п. Ячейка, в которой находится табличный курсор, является **активной**.

Курсор может принимать различный вид в зависимости от выполняемой операции:

	Указатель выделения	Для выделения одной ячейки, блока ячеек
	Указатель перемещения	Для перемещения данных
	Указатель маркера автозаполнения	Для копирования данных

В электронных таблицах можно работать как с одной ячейкой, так и с группой смежных или несмежных ячеек, которые образуют **диапазон (блок, интервал)**. Диапазон идентифицируется именами левой верхней и правой нижней ячеек. Имена ячеек в диапазоне разделяются двоеточием (:), например, блок **D1:D3** включает в себя ячейки **D1, D2, D3**, блок **A1:B4** включает в себя ячейки **A1, A2, A3, A4, B1, B2, B3, B4**. Выделение несмежного диапазона ячеек производится с помощью мыши при нажатой клавише **CTRL**, диапазоны несмежных ячеек разделяются точкой с запятой (;), например (**D1:D3; A1:B4**).

**Диаграмма** представляет собой графическое изображение связей между числами ЭТ.

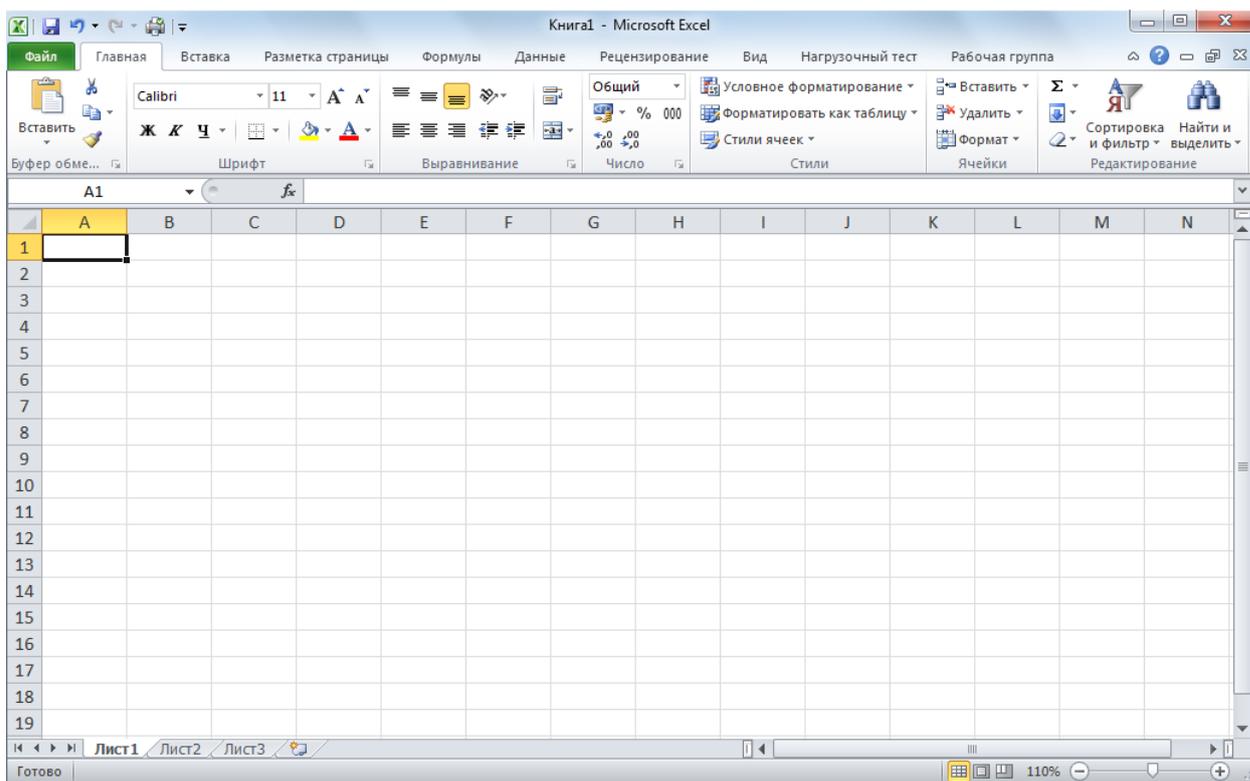
**Макротаблица** позволяет пользователю автоматизировать часто встречающиеся в повседневной работе операции.

#### ЗАПУСК ПРОГРАММЫ И СТРУКТУРА ОКНА EXCEL

После включения компьютера и загрузки операционной системы запустить программу MicrosoftExcel можно одним из следующих способов:

1. Двойной щелчок мыши по ярлыку программы MExcel на Рабочем столе.
2. При помощи команды в Главном меню (**ПУСК**→ **ПРОГРАММЫ**→ **MicrosoftExcel**).
3. Открытие документа, ранее созданного с помощью программы Excel.

После запуска программы на экран монитора будет выведено главное окно программы, в котором отображается рабочий лист **Лист1** рабочей книги. Структура экрана при работе с табличным процессором во многом походит на структуру экрана при работе с другими приложениями Windows.



В главном окне Excel содержатся:

1. Строка заголовка с кнопками управления окном программы и окном документа;
2. Строка меню, в которой содержатся команды меню программы;
3. Панели инструментов, которая предназначена для быстрого доступа к часто используемым командам и настройкам;
4. Строка формул (ввода данных), предназначенная для ввода и редактирования содержимого активной ячейки и поле имени, в которой указан адрес активной ячейки или размер выделенной области;
5. Рабочая область;
6. Ярлыки рабочих листов и кнопки навигации по ним;
7. Полосы прокрутки;
8. Строка состояния, отображающая сведения о выбранной команде или выполняемой операции, о включенных функциональных клавишах.

Как правило, при каждом запуске автоматически загружается чистая рабочая книга, состоящая из нескольких рабочих листов.

#### ЭТАПЫ ПРОЕКТИРОВАНИЯ ЭТ

Любая ЭТ состоит из следующих элементов:

1. Заголовка таблицы;
2. Заголовков столбцов («шапки» таблицы);
3. Информационной части (исходных и выходных данных, расположенных в соответствующих ячейках).

Процесс проектирования ЭТ включает следующие шаги:

1. Ввод заголовка ЭТ;
2. Ввод названий граф документа;
3. Ввод исходных данных;

4. Ввод расчетных формул;
5. Форматирование ЭТ с целью придания ей профессионального вида;
6. Подготовку ЭТ к печати и ее печать.

#### ВВОД ДАННЫХ В ТАБЛИЦУ

Ячейки имеют заданный формат, который можно установить с помощью команды меню **Главная**→**Ячейки** →**Формат**.

Для ввода данных необходимо:

1. Щелчком левой кнопки мыши активизировать нужную ячейку;
2. Набрать на клавиатуре нужную строку данных, при этом данные будут отображаться как в активной ячейке, так и в строке ввода данных;
3. Ввод данных в ячейку завершается нажатием клавиши **ENTER**, щелчком мыши вне рабочего пространства активной ячейки или щелчком по кнопке с изображением зеленой галочки в строке формул.

В ЭТ можно вводить и обрабатывать следующие типы данных:

- **ТЕКСТ** – любая последовательность символов, состоящая из цифр, пробелов и нецифровых символов и используемая для заголовков и комментариев. Введенный текст выравнивается в ячейке по левому нижнему краю.

- **ЧИСЛА** – числовая константа. Вводится в привычном виде со знаком или без него, в качестве разделителя целой и дробной части, может быть, точка или запятая в соответствии с настройками Windows. Введенные числа выравниваются в ячейке по правому краю. Для представления чисел по умолчанию электронные таблицы используют *Числовой* формат, который отображает два десятичных знака после запятой (например, **189,99**). *Экспоненциальный* формат применяется, если число, содержащее большое количество разрядов, не помещается в ячейке (например, число 2 000 000 000 в экспоненциальном формате будет записано: **2,00E+09**).

- **ДАТА И ВРЕМЯ** – символьная строка определенного формата, с которой можно выполнять арифметические и логические операции. Введенная дата интерпретируется как число и выравнивается по правому нижнему краю ячейки. В качестве разделителя используются следующие символы:

- точка 27.03.06 или 27.03.2006;
- дефис 27-03-06 или 26-мар-2006;
- косая черта 27/03/06 или 27/03/2006.

Дату и время можно также вводить при помощи функции с категорией *Дата и время*.

- **ФОРМУЛА** – выражение, состоящее из числовых величин и арифметических, логических и символьных операций. Формула может вводиться в одну или несколько ячеек и состоять из констант (числовых или текстовых), переменных, знаков математических операций и функций. Значения в функциях вычисляются *слева направо* с учетом скобок и

приоритета математических операций. При вводе формулы в ячейке отображается не сама формула, а результат вычислений по этой формуле.

#### АДРЕСАЦИЯ В EXCEL

**Ссылка** (адрес) на ячейку или диапазон ячеек указывает сведения о расположении значений или данных, которые требуется использовать в формуле. При помощи ссылок можно использовать значение одной и той же ячейки в нескольких формулах, а также можно сослаться на ячейки, находящиеся на других листах книги, в другой книге (внешние ссылки) или в другом приложении (удаленные ссылки). Для указания ссылки на ячейку вводится буква заголовка столбца и номер строки.

Адресация может быть **относительной**, **абсолютной** и **смешанной**. По умолчанию ссылки на ячейки в формулах рассматриваются как **относительные**, то есть при копировании формул адреса в ссылках автоматически **изменяются** в соответствии с относительным расположением исходной ячейки и создаваемой копии.

**Абсолютная** ссылка ячейки в формуле всегда ссылается на ячейку, расположенную в определенном месте. При **абсолютной** адресации адреса ссылок при копировании **не изменяются**. Элементы номера ячейки, использующие абсолютную адресацию, предваряются символом **\$**.

**\$A\$1** – абсолютная по столбцу и строке (ячейке).

**Смешанная** ссылка содержит либо абсолютный столбец и относительную строку, либо абсолютную строку и относительный столбец. При изменении позиции ячейки, содержащей формулу, относительная ссылка изменяется, а абсолютная ссылка не изменяется.

**\$A1** – смешанная с абсолютным столбцом;

**A\$1** – смешанная с абсолютной строкой.

Для изменения способа адресации необходимо выделить ссылку на ячейку и нажать клавишу **F4**.

#### ИСПОЛЬЗОВАНИЕ ФОРМУЛ ПРИ ВЫЧИСЛЕНИЯХ

Для выполнения вычислений **Excel** предлагает специальные средства расчетов – формулы и функции, которые позволяют автоматизировать вычисления. При вводе формул необходимо придерживаться следующих правил:

1. Формула должна начинаться со знака равенства (=) или плюс (+);
2. В формуле не должно быть пробелов;
3. При вводе формулы используются символы латинского алфавита;
4. В формулу, кроме числовых величин могут входить адреса ячеек и блоков, функции и другие формулы;
5. Длина формулы не может быть более 1024 символов;
6. Функция не может иметь более 30 аргументов;
7. В формуле можно использовать знаки арифметических операций: + (сложение), - (вычитание), / (деление), \* (умножение), ^ (возведение в степень), % (взятие процента);
8. В ячейке виден результат вычислений, сама формула отражается в строке ввода;

9. Формулу в ячейке можно редактировать в строке формул или непосредственно в ячейке, нажав клавишу **F2** или двойной щелчок мыши в ячейке;

10. Формула автоматически пересчитывается при каждом изменении содержимого таблицы.

**Примеры формул:**      =A1+100      =(B5\*C10)/100\*25

**Операторами** обозначаются операции, которые следует выполнить над операндами формулы (числа, логические значения, ссылки на адреса ячеек, встроенные функции). В Excel включено четыре вида операторов: арифметические, текстовые, адресные и операторы сравнения.

**Арифметические операторы** используются для выполнения основных математических вычислений над числами.

**Текстовый оператор** & (амперсant) используется для обозначения операции объединения последовательностей символов в единую последовательность. Например, результатом выполнения выражения «Северный» & «ветер» будет: «Северный ветер».

**Операторы сравнения** используются для обозначения операций сравнения двух чисел. Результатом выполнения операции сравнения является логическое значение **ИСТИНА** или **ЛОЖЬ** (>=, >, =).

**Адресные операторы** объединяют диапазоны ячеек для осуществления вычислений (двоеточие, запятая и пробел).

Табличный процессор **Excel** содержит большое количество стандартных формул, называемых **функциями**. Функции используются для простых или сложных вычислений. Вызов функции состоит в указании в формуле имени функции, после которого в скобках указывается список аргументов, разделенных точкой с запятой (;). В качестве аргумента может использоваться число, адрес ячейки или выражение. Excel содержит функции следующих типов:

1. Финансовые
2. Функции даты и времени
3. Математические и тригонометрические
4. Статистические
5. Функции просмотра и ссылок
6. Функции для работы с базой данных
7. Текстовые функции
8. Логические функции
9. Информационные функции
10. Инженерные

Ввод функций можно осуществлять несколькими способами:

- а. Ручным набором имени функции и её аргументов;
- б. Выбором из списка функций с помощью команды **Формулы** → **Вставить функцию**;
- в. Выбором функции щелчком по значку  на строке формул;

г. Кнопка **Автосумма** ( $\Sigma$ ) на вкладке **Главная**.  
Рассмотрим некоторые типы функций для анализа данных:

### 1. СТАТИСТИЧЕСКИЕ ФУНКЦИИ

Функция	Назначение
МАКС	Возвращает наибольшее значение из списка аргументов
МИН	Возвращает наименьшее значение из списка аргументов
СРЗНАЧ	Возвращает среднее арифметическое аргументов
СЧЁТ	Подсчитывает количество чисел в списке аргументов

### 2. МАТЕМАТИЧЕСКИЕ ФУНКЦИИ

Функция	Назначение
СУММ	Суммирует числа
СТЕПЕНЬ	Возвращает результат возведения числа в степень
СУММПРОИЗВ	Возвращает сумму произведений соответствующих элементов массивов или диапазонов

**Редактирование таблицы** – это внесение каких-либо изменений в существующую таблицу.

Для исправления ошибок в таблице можно использовать способы:

1. Набор новой информации поверх ошибочной;
2. Активизация строки ввода;
3. Двойной щелчок мыши по ячейке (или клавиша **F2**).

Для удаления содержимого ячеек используют клавишу **DELETE**

Перемещение в таблице:

Клавиши со стрелками (←↑→↓)	Переход на одну ячейку вверх, вниз, влево или вправо
HOME	Переход в начало строки
CTRL+HOME	Переход в начало листа
CTRL+END	Переход в последнюю ячейку на листе крайнего справа используемого столбца
PAGE DOWN	Переход на один экран вниз
PAGE UP	Переход на один экран вверх

Выделения всей строки целиком - щелчок левой кнопкой мышью по заголовку строки или **SHIFT+«ПРОБЕЛ»**.

Выделения всего столбца целиком - щелчок левой кнопкой мышью на его заголовке или **CTRL+«ПРОБЕЛ»**.

Выделения всего рабочего листа - щелчок левой кнопкой мыши на пересечении полос с именами столбцов и строк или **CTRL+SHIFT+«ПРОБЕЛ»**. **ФОРМАТИРОВАНИЕ ТАБЛИЦ**

Под **форматированием** таблицы понимают изменение её внешнего вида. Внешний вид таблицы не отражается на ее функциональных свойствах, но дополнительное форматирование может значительно улучшить восприятие данных и облегчить работу с ними. Существуют следующие способы изменения внешнего вида таблиц:

1. Форматирование текста, чисел, вставка пробелов и выравнивание содержимого ячеек для улучшения визуального восприятия;

2. Форматирование цветов, палитр и границ ячеек для повышения наглядности;

3. Копирование форматов ячеек и возможность сохранения стилей форматирования для ускорения процесса форматирования;

4. Применение автоформатирования для достижения наибольшей эффективности в работе.

Перед выполнением любой команды форматирования следует выделить ячейку или область, на которую будет распространяться действие команды. Если область не выделена, параметры будут задаваться для активной ячейки.

С помощью вкладки **Главная**→**Шрифт**, **Выравнивание** или **Число** можно изменить шрифт, размер символов, выравнивание, стиль и цвет содержимого выделенных ячеек.

Чтобы получить доступ ко всем возможностям форматирования Excel необходимо воспользоваться командой **Главная**→**Ячейки**→**Формат**→**Формат ячеек** или при помощи команды контекстного меню **Формат ячеек...** В результате появится диалоговое окно **Формат ячеек** с вкладками, которое поможет изменить вид любой ячейки или диапазона ячеек:

Вкладка	Возможность форматирования
Число	Можно выбрать любой числовой формат, при помощи которого можно задать число дробных десятичных знаков, а также использование знака валюты или символа процентов, форматирование дат и текста
Выравнивание	Для выравнивания содержимого выбранной ячейки (по умолчанию числа выравниваются по правому краю ячейки, а текст – по левому). Можно также задать направление текста, разрешить перенос длинных строк по словам ( <b>ALT+ENTER</b> ), объединить ячейки и произвести автоподбор ширины.
Шрифт	Можно определить шрифт, размер, начертание, цвет, подчеркивание (или зачеркивание) для каждого символа в ячейке.
Граница	Можно задать использование границ для выделения итогов или обозначения диапазонов. Можно выбрать тип и цвет линии. В отличие от линий сетки линии обрамления будут видны на печати.
Заливка	Можно задать иной цвет фона ячейки или применение узора.
Защита	Можно защитить содержимое выделенной ячейки (ячеек), чтобы скрыть формулы или предотвратить случайное изменение содержимого.

Отображаемая ширина столбца показывает среднее число знаков цифр 0 - 9 стандартного шрифта, помещающихся в ячейке. Ширину столбца в Excel можно изменить только для всех ячеек данного столбца. Так же и высоту строки можно изменить только для всех ячеек данной строки. Этим достигается однородность при построении таблицы.

Ширину столбца и высоту строки можно изменить разными способами:

- с помощью указателя мыши (, ) – подвести указатель к правой границе имени столбца или нижней границе номера строки, чтобы он принял вид двунаправленной стрелки и при нажатой кнопке мыши перетащить границу в нужном направлении;

- двойным щелчком мыши по правой границе столбца в области имени или нижней границе номера строки – при этом ширина столбца установится в соответствии с самой длинной строкой данных в столбце, а высота строки преобразуется по высоте самого высокого символа в этой строке;

- с помощью команды **Главная→Ячейки→Формат → Ширина столбца (Высота строки)**.

Если выделены несколько столбцов или строк, перетаскивание правой границы самого правого столбца или нижней границы строки из выделенных приведет к изменению размеров всех выделенных столбцов или строк.

#### ПОСТРОЕНИЕ ДИАГРАММ И ГРАФИКОВ

В MSExcel имеется возможность графического представления данных в виде диаграммы. **Диаграмма** – это условное изображение числовых величин или их соотношений графическим способом. Диаграммы связаны с данными листа, на основе которых они были созданы, и изменяются каждый раз, когда изменяются данные на листе.

Диаграммы могут использовать данные смежных и несмежных ячеек. Для создания диаграммы из несмежных ячеек нужно выделить первую группу ячеек, содержащих необходимые данные, далее, удерживая клавишу **CTRL**, выделить необходимые дополнительные группы ячеек.

Можно создать либо внедренную диаграмму, либо лист диаграммы. **Внедренная диаграмма** – это объект, расположенный на листе с данными, по которым она построена. Её можно перемещать по экрану и изменять её размеры. **Лист диаграммы** – новый рабочий лист, содержащий только диаграмму.

Для построения диаграмм используют **Мастер диаграмм**, запускаемый командой **Вставка→Диаграммы**. На первом этапе необходимо выбрать тип диаграммы и её вид. Второй этап служит для выбора данных, по которым будет построена диаграмма. Если диапазон был выбран заранее, то в области предварительного просмотра появится приблизительное отображение будущей диаграммы. Третий этап работы мастера состоит в выборе оформления диаграммы. На вкладках окна мастера задаются:

1. Название диаграммы, подписи осей (Заголовки);
2. Отображение и маркировка осей координат (Оси);
3. Отображение сетки линий (Линии сетки);
4. Описание построенных графиков (Легенда);
5. Отображение надписей, соответствующих отдельным элементам данных на графике (Подписи данных);
6. Представление данных, использованных при построении графика, в виде таблицы (Таблица данных).

В зависимости от типа диаграммы некоторые вкладки могут отсутствовать.

#### **Структура диаграммы:**

1. Область диаграммы;

2. Область построения диаграммы;
3. Ряд данных;
4. Ось значений и ось категорий;
5. Основные линии сетки;
6. Легенда.

Готовую диаграмму можно изменить. Для этого щелкнуть по редактируемому элементу диаграммы. Элемент выделяется маркерами, для его форматирования можно воспользоваться помощью вкладок **Конструктор, Макет, Формат** (область **Работа с диаграммами**).

#### СОРТИРОВКА ДАННЫХ

Упорядочивание значений диапазона ячеек в определённой последовательности называется **сортировкой**. В Excel можно сортировать как числовые, так и текстовые данные. При этом текстовые данные будут сортироваться в алфавитном или в обратном алфавитном порядке, а числовые данные – в порядке возрастания или убывания в зависимости от заданного порядка сортировки.

Для сортировки данных используют команду **Главная→Редактирование→Сортировка и фильтр**.

Пользователь может задать три уровня сортировки одновременно. Сначала сортировка будет выполнена по первому уровню, затем в полученном списке – по второму, а потом – по третьему.

#### АВТОМАТИЗАЦИЯ ВВОДА

Часто таблицы содержат повторяющиеся или однотипные данные, для ввода таких данных Excel имеет средства автоматизации: автозавершение, автозаполнение числами, автозаполнение формулами.

Автозавершение используется для автоматизации ввода текстовых данных. Заполняемые ячейки должны идти подряд, без промежутков. Программа сравнивает вводимые символы с ранее введенными строками и при обнаружении совпадения автоматически дополняет текст. Для подтверждения автозавершения – нажать клавишу **ENTER**, если автозавершение не нужно – продолжить набор. Прервать работу средства автозавершения можно оставив в столбце пустую ячейку.

При активировании ячейки в правом нижнем углу рамки появляется черный квадратик – маркер заполнения. Перетаскивание маркера заполнения рассматривается как операция «размножения» содержимого ячейки в горизонтальном или вертикальном направлении. При перетаскивании маркера с помощью левой кнопки мыши происходит копирование содержимого исходной ячейки. При перетаскивании с помощью правой кнопки – можно выбрать способ автозаполнения:

- копировать ячейки – все ячейки будут содержать исходную информацию;
- заполнить – ячейки будут заполнены арифметической прогрессией.

Функция ввода прогрессий позволяет оперативно создавать последовательности из возрастающих (убывающих) по арифметическому ил геометрическому закону чисел, дат и т.п.

Для создания *прогрессии* необходимо:

1. Ввести значение первого элемента прогрессии в выбранную базовую ячейку.
2. Выделить блок ячеек, который должны будут занять члены прогрессии.
3. Выполнить команду **Главная** → группа **Редактирование** → **Заполнить** → **Прогрессия**,
4. Указать тип и параметры формируемой последовательности значений.

Автозаполнение формулами выполняется аналогично автозаполнению числами. При автозаполнении формулами относительные ссылки изменяются в соответствии с относительным расположением копии и оригинала, абсолютные остаются без изменений. Правила обновления ссылок при автозаполнении формулами приведены в таблице.

Таблица

Ссылка в исходной ячейке	Ссылка в следующей ячейке	
	При заполнении вправо	При заполнении вниз
A1 (относительная)	<b>B1</b>	<b>A2</b>
\$A1 (абсолютная по столбцу)	\$A1	\$A2
A\$1 (абсолютная по строке)	B\$1	A\$1
\$A\$1 (абсолютная)	\$A\$1	\$A\$1

## 5.5. Создание презентаций с помощью программы PowerPoint

Включённая в состав офисного пакета Microsoft Office, программа Microsoft Office PowerPoint является простым в освоении и очень мощным инструментом создания привлекательных презентаций, отвечающих любым требованиям. С помощью презентации PowerPoint каждый при желании может организовать эффективное сопровождение своего выступления.

Формы проведения презентации PowerPoint могут быть очень разнообразны: выступление, доклад, демонстрация кинофильма или фотографий, выставка или даже театрализованное представление. Презентация может быть простая, схематичная, она может содержать графики и схемы, множество иллюстраций, изобилловать спецэффектами. Все зависит от того, насколько развита фантазия и умения у её создателя.

**Презентация** – документ, содержащий ряд кадров (слайдов). Важным понятием презентации является слайд.

**Слайд** – логически и художественно завершённый документ. Презентация является мультимедийным документом, т.к. каждый слайд может включать в себя различные формы представления информации (текст, таблицы, диаграммы, изображения, звук, анимацию и др.).

### РЕЖИМЫ РАБОТЫ ПРОГРАММЫ

В приложении PowerPoint существует четыре основных режима просмотра слайдов: *обычный, сортировщик слайдов, режим чтения и показа слайдов.*

Режимы представляют собой разные способы отображения слайдов на экране. В зависимости от установленного режима изменяется вид окна презентации и компоновка его базовых элементов. Для переключения режима просмотра слайдов можно использовать кнопки режимов просмотра слайдов в строке состояния или команду **Вид** в группе **Режимы просмотра презентации**. По умолчанию, установлен режим просмотра слайдов **Обычный** (рис.1).

**Обычный режим** используется при создании и редактировании презентаций. В данном режиме на экране монитора по умолчанию появляются три области.

Первая область – панель слайдов с вкладками **Слайды** и **Структура** располагается слева и используется для редактирования структуры презентаций. В этом режиме на экран выводятся только заголовки и подзаголовки слайдов.

Вторая область – область редактирования слайда размещена на экране справа. В этом месте в крупном масштабе отображается текущий слайд.

Третья область – заметки к слайду используется для хранения заметок к слайду. Она расположена на экране снизу. Заметки предназначены для докладчика и не видны во время демонстрации слайдов.

**Режим сортировщика слайдов** даёт возможность видеть одновременно несколько десятков слайдов, выполнить реорганизацию

презентации, провести предварительный просмотр эффектов анимации и смены слайдов. Под каждым слайдом указан его порядковый номер

**В режиме чтения слайдов** презентацию можно посмотреть в окончательном виде на полном экране. Режим чтения предназначен только для демонстрации готовой презентации на экране

По умолчанию в **режиме показа** переход к следующему слайду осуществляется щелчком мыши, а перемещение вперед и назад – клавишами управления курсором или PageUp / PageDown. Для выхода из режима показа слайдов нужно нажать клавишу Esc или ввести команду контекстного меню **Завершить показ слайдов**. В этом режиме нельзя осуществлять действия по редактированию слайдов, добавлению в презентацию какого-либо материала, настройке параметров внешнего вида и т. д. Чтобы просмотреть презентацию, выберите команду **Показ слайдов→С начала** или **С текущего слайда**.

Если вы хотите, чтобы указатель мыши был не виден во время презентации, щелкните во время показа правой кнопкой мыши по экрану, выберите **Указатель→Параметры стрелки→Скрывать всегда** (для возврата указателя выберите команду **Видимый** или **Автоматически**).

Если вы хотите выключить экран, щелкните правой кнопкой мыши по экрану, выполните **Экран → Черный экран**, щелчок мыши выведет на экран следующий слайд.

Если вы хотите графически отметить некоторые моменты, так чтобы не сохранять пометки, щелкните правой кнопкой мыши, выберите **Указатель→Карандаш**, цвет карандаша можно выбрать, выполнив **Указатель →Цвет чернил**.

#### СПОСОБЫ СОЗДАНИЯ ПРЕЗЕНТАЦИИ

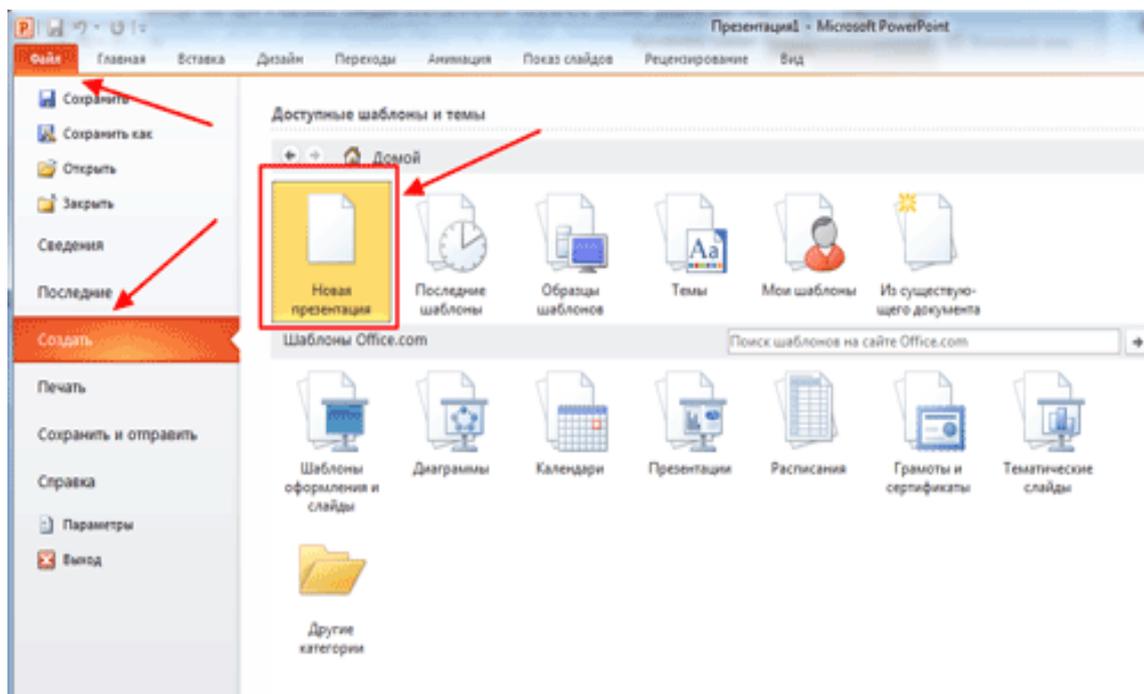
Программа PowerPoint обладает всеми необходимыми инструментами, позволяющими за короткое время создавать профессиональные слайды.

Запуск программы можно осуществить одним из следующих способов:

- Команда **Пуск→Все программы→MSOffice→MSPowerPoint 2010**.
- Двойной щелчок мыши по ярлыку программы на Рабочем столе.
- Открытие документа, ранее созданного с помощью программы **PowerPoint**.

При запуске программы MS PowerPoint 2010 новая презентация создается автоматически. Новая презентация содержит только один титульный слайд.

Чтобы создать презентацию на базе шаблона **Новая презентация** откройте вкладку **Файл → Создать**, в области **Доступные шаблоны и темы** щелкните **Новая презентация** (Рис. 1).



**Рис. 1** – Создание новой презентации Microsoft PowerPoint 2010

При выборе **Новая презентация**, запускается пустая презентация, и предлагаются шаблоны для создания различных типов слайдов презентации, перечень которых располагается в пунктах **Доступные шаблоны и темы** и **Шаблоны Office.com**. Шаблоны для PowerPoint можно выбрать и с помощью команды **Установленные шаблоны**, где найдёте шаблоны «Классический фотоальбом», «Современный фотоальбом», «Рекламный буклет», «Викторина», «Широкоэкранный презентация».

Программа PowerPoint предлагает шаблоны слайдов для: текста, текста с диаграммами, графики и других объектов. Если не подходит ни один из этих шаблонов, можно выбрать пустой слайд, на котором можно поместить текст, графику и другие элементы по своему усмотрению.

Каждый слайд презентации MS PowerPoint характеризуется следующими свойствами:

- разметкой слайда (расположением заголовков, текста и объектов на слайде);
- шаблоном оформления (дизайн слайда);
- эффектами перехода от слайда к слайду.

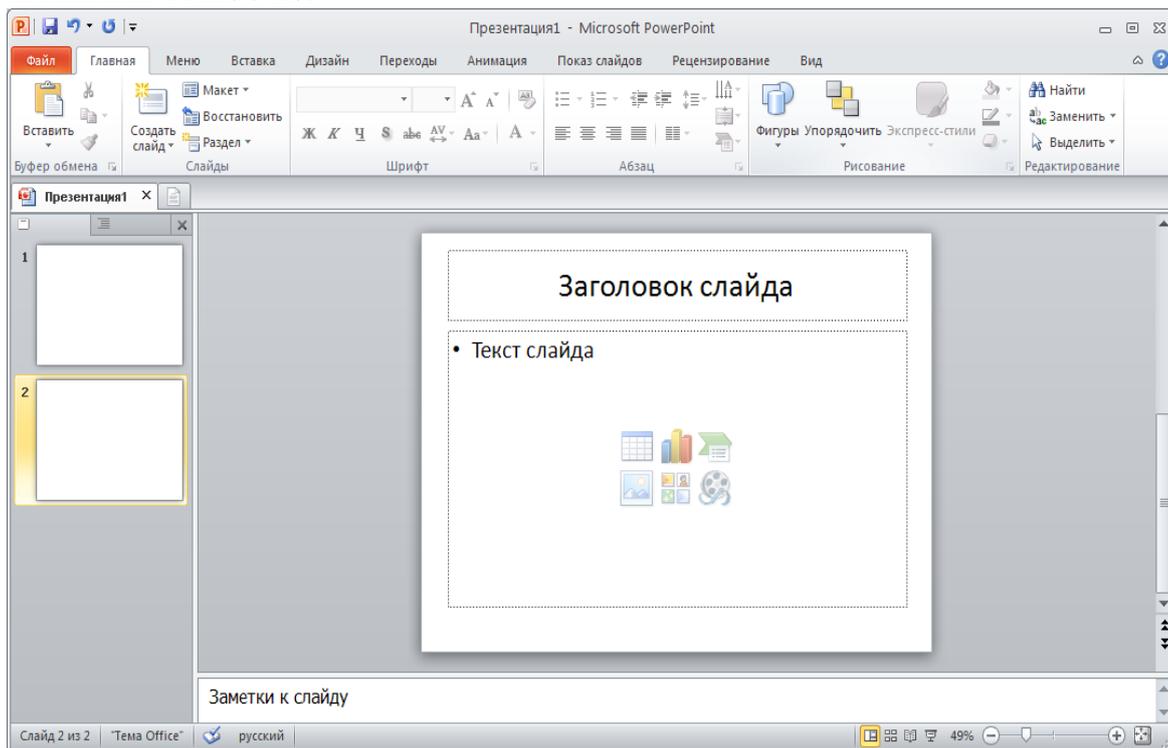
Эти свойства слайда влияют на его отображение во время демонстрации.

В одной презентации могут быть слайды различных типов. Созданный файл сохраняется по умолчанию с расширением \*.pptx.

#### ЭЛЕМЕНТЫ ОКНА POWERPOINT

После запуска программы на экран монитора будет выведено главное окно программы Microsoft PowerPoint (рис. 2), в котором отображается рабочее поле со слайдом. Структура окна программы имеет стандартный вид

и во многом походит на структуру экрана при работе с другими приложениями Windows.



**Рис. 2** - Рабочее окно программы PowerPoint

Основные элементы интерфейса программы PowerPoint 2010:

- строка заголовка окна, содержащая имя файла презентации;
- панель быстрого доступа, на которой располагаются часто используемые команды. Можно добавить или удалить кнопки на панели быстрого доступа, нажав кнопку ▼;
- лента с вкладками для доступа к элементам управления;
- область редактирования слайда;
- панель слайдов, на которой отображаются эскизы слайдов. На панели слайдов можно добавлять или удалять слайды, а также изменять порядок слайдов в презентации;
- панель **Заметки** служит для создания заметок к текущему слайду, которые можно просматривать в режиме докладчика во время демонстрации презентации;
- строка состояния, содержащая информацию о текущем слайде: номер слайда, тема, язык;
- кнопки режимов просмотра слайдов: обычный, сортировщик слайдов, режим чтения, показ слайдов;
- кнопки и ползунок для изменения масштаба слайда.

## РЕЖИМЫ ОБРАЗЦОВ СЛАЙДОВ

Образец слайдов занимает самую высокую позицию в иерархии слайдов и содержит сведения о теме и макетах слайдов в презентации, включая такие элементы оформления как фон, цвета, шрифты, эффекты, размер заполнителей и размещение.

В каждой презентации есть хотя бы один образец слайдов. Главное преимущество изменения и использования образцов слайдов заключается в том, что можно применить общее изменение стиля ко всем слайдам презентации, включая и те слайды, которые могут быть добавлены позже. Использование образца слайдов помогает экономить время, так как повторяющиеся данные достаточно ввести на одном слайде.

Режимы образцов слайдов находятся на вкладке **Вид** и включают представления: **образец слайдов, образец выдач и образец заметок**.

Вкладка **Образец слайдов** предназначена для изменения дизайна и макета образцов слайдов.

### ВСТАВКА ОБЪЕКТОВ НА СЛАЙД

Добавление объектов на слайд может осуществляться как с помощью команд ленты, так и с помощью заполнителей, размещенных в макетах слайдов. Все встроенные макеты слайдов (за исключением макета Пустой слайд) содержат заполнители содержимого. Текстовые заполнители отображаются в виде пунктирных прямоугольников с элементом заголовка или текста слайда. Щелчок по надписи активирует поле для ввода текста. Объектные заполнители отображаются в виде пунктирных прямоугольников со значками объектов (таблица, диаграмма, рисунок SmartArt, рисунок из файла, картинка, клип мультимедиа).

Если необходимо вставить в слайд изображение из файла, щелчок по значку **Вставить рисунок из файла** вызовет диалоговое окно для указания пути к файлу. Если необходимо добавить картинку, щелчок по значку **Картинка** откроет дополнительную область для поиска изображений, содержащихся в библиотеке клипов Microsoft Office.

Размеры любого объекта на слайде можно изменить, используя маркеры изменения размер.

Для добавления объектов на слайд используется вкладка **Вставка**.

### СОЗДАНИЕ ССЫЛОК В ПРЕЗЕНТАЦИИ

В приложении PowerPoint гиперссылка осуществляет связь одного слайда с другим в одной и той же презентации или со слайдом в другой презентации, адресом электронной почты, веб-страницей или файлом.

В программе PowerPoint в качестве гиперссылки может использоваться фрагмент текста или любое графическое изображение. Гиперссылку можно добавить к любому тексту или объекту на слайде: к фигурам, таблицам, диаграммам и рисункам.

Использование гиперссылок заметно упрощает перемещение по слайдам презентации во время показа, а также ускоряет доступ к необходимой информации из внешних источников.

Гиперссылки активны только в режиме **Показ слайдов**. Возможности программы PowerPoint не позволяют увидеть действие гиперссылки в других режимах просмотра презентации.

#### **Создание гиперссылки на слайд в той же презентации:**

- В **обычном режиме** просмотра выделите текст или объект, который нужно использовать как гиперссылку.
- На вкладке **Вставка** в группе **Связи** нажмите кнопку **Гиперссылка**, откроется диалоговое окно **Вставка гиперссылки**.
- В поле **Связать с** выберите пункт **местом в документе**.
- Выполните:
  - для ссылки на слайд в текущей презентации в поле **Выберите место в документе** щелкните слайд, который нужно использовать в качестве цели гиперссылки.
  - для ссылки на произвольный показ слайдов в текущей презентации в поле **Выберите место в документе** щелкните произвольный показ, который нужно использовать как цель гиперссылки, установите флажок **Показать и вернуться**.

#### **Создание гиперссылки на слайд в другой презентации:**

- В **обычном режиме** просмотра выделите текст или объект, который нужно использовать как гиперссылку.
- На вкладке **Вставка** в группе **Связи** нажмите кнопку **Гиперссылка**.
- В поле **Связать с** выберите пункт **файлом, веб-страницей**.
- Укажите презентацию, которая содержит целевой слайд гиперссылки. Щелкните пункт **Закладка**, затем щелкните **заголовок слайда**, на который будет указывать ссылка.

#### **Создание гиперссылки на страницу или файл в Интернете:**

- В **обычном режиме** просмотра выделите текст или объект, который нужно использовать как гиперссылку.
- На вкладке **Вставка** в группе **Связи** нажмите кнопку **Гиперссылка**.
- В поле **Связать с** выберите пункт **файлом, веб-страницей** и нажмите кнопку **Интернет**.
- Укажите путь и выберите страницу или файл, на которые будет указывать гиперссылка.

#### **ИСПОЛЬЗОВАНИЕ КОЛОНТИТУЛОВ**

Колонтитулы – это удобный способ размещения повторяющегося текста в верхней и нижней части каждого слайда, а также на страницах выдач и заметок. С помощью колонтитулов можно разместить на слайдах презентации такие данные, как дата, номер слайда, имя автора и т. д.

Добавление колонтитулов осуществляется с помощью кнопки **Колонтитулы** в группе **Текст** на вкладке **Вставка**. Диалоговое окно **Колонтитулы** содержит две вкладки: **Слайд** и **Заметки и выдачи**. Это

позволяет задавать разные колонтитулы для слайдов и страниц заметок и выдач. Созданные колонтитулы можно применить как для одного, так и для всех слайдов презентации. Можно убрать колонтитул с титульного листа презентации.

## 5.6. Вопросы для самопроверки по теме 5

*Задание №1.* Вид документа в MS Word полностью совпадает с его представлением для печати в режиме:

1. Обычный;
2. Веб-документ;
3. Разметка страницы;
4. Структура.

*Задание №2.* К операциям редактирования текста относятся:

1. форматирование символов;
2. перемещение фрагмента текста;
3. копирование фрагмента текста;
4. проверка правописания;
5. определение абзацного отступа.

*Задание №3.* Для автоматического создания оглавления в документе MS Word необходимо выполнить следующее условие:

1. страницы должны быть автоматически пронумерованы;
2. все пункты должны быть одинаково отформатированы;
3. все пункты и подпункты должны быть отформатированы с использованием стилей «Заголовки» различных уровней;
4. все пункты должны являться списком (нумерованным, маркированным, многоуровневым).

*Задание №4.* В ячейке B2 рабочего листа MS Excel содержится формула =D1+\$A\$1, эта формула скопирована в ячейку C3. Какая формула будет содержаться в ячейке C3?

1. =D1+\$A\$1;
2. =D1+\$B\$2;
3. =E2+\$A\$1;
4. =E2+\$B\$2.

*Задание №5.*

Дана таблица:

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>1</b>	2	4	6	=СУММ(A1:C1)
<b>2</b>	1	5	4	=МАКС(A1:C2)
<b>3</b>				=D1+D2

Формула в ячейке D3 после произведенных расчетов имеет значение:

1. 13
2. 17
3. 18
4. 22

*Задание №6.* Упорядочивание значений диапазона ячеек в определенной последовательности называется:

1. форматирование;
2. фильтрация;
3. группировка;
4. сортировка.

*Задание №7.* Последовательность слайдов, содержащих мультимедийные объекты, называется:

1. иерархические объекты;
2. гипермедиа-ссылки;
3. презентация;
4. фрейм.

*Задание №8.* Файл, созданный в программе MS Power Point, автоматически получает расширение:

1. pptx;
2. wmf;
3. ffa;
4. prv

## 6. ОСНОВНЫЕ ПОНЯТИЯ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ

### 6.1. Базы данных и системы управления ими: основные понятия и определения, реляционная модель данных.

Для обеспечения эффективности информационных процессов необходима соответствующая организация данных.

В настоящее время в самых различных областях широкое распространение получила разработка баз данных (БД).

**База данных (БД)** – это совокупность взаимосвязанных групп данных, предназначенных для обработки на ЭВМ, в которых информация структурирована таким образом, чтобы их могли использовать различные пользователи и программы.

**Система базы данных** состоит из базы данных и программного обеспечения общего назначения, называемого **Системой управления базами данных (СУБД)**, служащего для управления базой данных.

**СУБД** – это комплекс программных средств, предназначенных для создания структуры базы данных, наполнения, хранения, извлечения, поиска, редактирования её содержимого и визуализации информации.

Основные требования к СУБД:

1. **Непротиворечивость данных.** Поддержание полной, непротиворечивой и адекватно отражающей предметную область информации.

2. **Актуальность хранимых данных.** В любой момент времени информация, содержащаяся в БД, должна быть современной.

3. **Многоаспектное использование данных.** Поступление информации из различных источников в единую БД возможность её использования.

4. **Возможность модификации системы** – возможность расширения, дополнения новыми функциями и модификации данных без ущерба для системы в целом.

5. **Надежность** – целостность БД не должна нарушаться при технических сбоях.

6. **Скорость доступа** – обеспечение быстрого доступа к требуемой информации.

СУБД осуществляют взаимодействие между БД и пользователями системы, а также между БД и прикладными программами, реализующими определенные функции обработки данных.

СУБД обеспечивают надежное хранение больших объемов данных сложной структуры во внешней памяти компьютера и эффективный доступ к ним.

К основным функциям СУБД относятся:

1. Непосредственное управление данными во внешней и оперативной памяти и обеспечение эффективного доступа к ним в процессе решения задач.

2. Поддержание целостности данных.
3. Ведение системного журнала изменений в БД для обеспечения восстановления БД после технического или программного сбоя.
4. Реализация поддержки языка описания данных и языка запросов.
5. Обеспечение безопасности данных.
6. Обеспечение параллельного доступа к данным нескольких пользователей.

Реализация информационных моделей осуществляется в рамках СУБД. Каждая конкретная СУБД работает с определенной **моделью данных**.

**Модель данных** представляет собой множество структур данных и операций над этими структурами. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных должна обеспечивать:

1. Допустимую структуру данных для описания разнообразных объектов;
2. Множество допустимых операций над данными;
3. Ограничение контроля целостности (логические ограничения модели на данные для сохранения непротиворечивости данных и адекватного отображения предметной области в БД).

Различают три основных типа **моделей данных**:

1. **Иерархическая** (древовидная структура), существует главный объект и подчиненные объекты.
2. **Сетевая**, любой объект может быть главным и подчиненным.
3. **Реляционная**, данные и взаимосвязи между ними представлены в виде таблиц.

**Реляционная модель** данных является одной из самых распространенных моделей, используемых в современных СУБД

К **достоинствам** реляционной модели относятся:

1. Простота и доступность для понимания пользователем. Информационной конструкцией модели является таблица.
2. При проектировании применяются строгие правила, базирующиеся на математическом аппарате.
3. Полная независимость данных. При изменении структуры реляционной БД изменения, которые требуется произвести в прикладных программах, минимальны.
4. Для построения запросов и написания прикладных программ нет необходимости знания конкретной организации БД во внешней памяти.

**Недостатки** реляционной модели:

1. Имеет более низкую скорость доступа и требует большего объема внешней памяти.
2. Появление большого количества таблиц в результате проектирования затрудняет понимание структуры данных.
3. Не всегда предметную область можно представить в виде совокупности таблиц.

**Реляционная модель** ориентирована на организацию данных в виде **двумерной таблицы**. Столбцы такой таблицы называют **полями**, каждое поле характеризуется своим именем и типом данных, представляющих значения данного свойства. Строки таблицы являются **записями** об объекте, эти записи разбиты на поля столбцами таблицы, поэтому каждая запись представляет собой набор значений, содержащихся в полях (рис.1).

	Поле 1	Поле 2	...	Поле n
Запись 1				
Запись 2				
...				
Запись m				

Рис. 1. Структура данных реляционной модели

Каждая реляционная таблица обладает следующими **свойствами**:

1. Структура таблицы определяется совокупностью столбцов;
2. В каждой строке таблицы содержится по одному значению в соответствующем столбце;
3. Одинаковые строки в таблице отсутствуют;
4. Общее число строк не ограничено;
5. Столбцы таблицы имеют уникальные имена;
6. Значения каждого столбца относятся к одному и тому же типу;
7. Порядок следования строк и столбцов является произвольным.

Каждая таблица должна содержать, по крайней мере, одно **ключевое поле**, содержимое которого уникально для каждой записи в таблице.

**Ключевое поле** – это столбец (может быть несколько столбцов), добавляемый к таблице и позволяющий установить связь с записями в другой таблице. Существуют ключи двух типов: **первичные** и **вторичные (внешние)**.

**Первичный ключ** - это одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ не допускает значений **Null** (значение, которое можно ввести в поле или использовать для указания отсутствующих или неизвестных данных) и всегда должен иметь уникальный индекс (не допускается дублирование ключа).

Существует три типа первичных ключей: **ключевые поля счетчика** (счетчик), **простой** ключ и **составной** ключ.

**Поле счетчика** (Тип данных «Счетчик»), в котором для каждой добавляемой в таблицу записи в поле автоматически заносится уникальное числовое значение.

Если первичный ключ содержит уникальные значения и состоит из одного поля (например, коды или инвентарные номера), он является **простым**.

Первичный ключ является **составным**, если он состоит из нескольких полей и невозможно гарантировать уникальность значений каждого поля.

Чаще всего такая ситуация возникает для таблицы, используемой для связывания двух таблиц отношением многие-ко-многим.

Если возникают затруднения с выбором подходящего типа первичного ключа, то в качестве ключа целесообразно выбрать **поле счетчика**.

Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.

**Внешний (вторичный) ключ** - это одно или несколько полей (столбцов) в таблице, содержащих ссылку на поле (поля) первичного ключа в другой таблице, т.е. не является уникальным. Служит для организации связей между таблицами и поддержания целостности и непротиворечивости информации в базе данных.

Из двух логически связанных таблиц одну называют таблицей первичного ключа или **главной** таблицей, а другую таблицей вторичного (внешнего) ключа или **подчиненной** таблицей.

**Реляционная база данных (РБД)** – это совокупность реляционных таблиц, логически взаимосвязанных и отражающих некоторую предметную область.

В настоящее время в мире насчитывается более 50 типов разнообразных по своим функциональным возможностям СУБД для IBM PC – совместимых компьютеров. К числу современных реляционных баз данных относятся Paradox, MSSQLServer, Oracle, MS ACCESS и др.

## **6.2. СУБД MS Access, основные этапы разработки базы данных**

### **ОБЪЕКТЫ MS СУБД**

**MS ACCESS** - это 32-разрядная система управления базами данных (СУБД), предназначенная для работы на персональном компьютере (ПК) или локальной вычислительной сети (ЛВС) под управлением семейства операционных систем (ОС) MicrosoftWindows.

Основные характеристики **СУБД MSAccess**:

1. Графический многооконный интерфейс, позволяющий в диалоговом режиме создавать таблицы, формы, запросы, отчеты и макросы;
2. Наличие развитых диалоговых средств конструирования;
3. Наличие специальных инструментальных средств - мастеров;
4. Наличие средств соблюдения целостности данных на уровне базы данных;
5. Наличие средств обеспечения безопасности данных (защита паролем, ограничение уровня доступа и т. д.)
6. Возможность работы в многопользовательских системах (WindowsNT, NovellNetWare);
7. Использование объектной технологии OLE для внедрения в базу данных различных объектов (тексты, электронные таблицы, рисунки и т.д.);
8. Наличие собственного языка программирования VisualBasic — объектно-ориентированного языка для разработки приложений пользователя;
9. Полная поддержка языка запросов SQL;

**Access. Создание запросов, форм и отчетов.**

**СУБД MS Access** ориентирована на работу с **объектами**, предназначенными для ввода, управления и хранения информации. Все объекты одной БД хранятся в одном и том же файле, имеющем оригинальное имя и стандартное расширение \*.accdb (раньше – \*.mdb. Этими объектами являются:

- **Таблицы** – это основа базы данных, предназначены для хранения информации по объектам модели данных предметной области.

- **Запросы** – это средство выборки необходимых данных по различным критериям из одной или нескольких таблиц БД. Запрос может формироваться с помощью запросов по образцу (QBE) или с помощью языка структурированных запросов (SQL).

- **Формы** – это электронный вариант физических форм документов. Они предназначены для ввода, просмотра и редактирования данных, могут использоваться для создания панелей управления в приложениях пользователя.

- **Отчеты** - это средство, которое используется для формирования выходных документов, предназначенных для вывода на экран, принтер или в файл, а также группировки, сортировки, определения итоговых значений.

- **Макросы** – это последовательность макрокоманд для расширения возможностей СУБД. С их помощью можно изменять ход выполнения приложения, открывать, фильтровать и изменять данные в формах, отчетах, выполнять запросы и создавать новые таблицы.

- **Модули** – это объект, содержащий программы на языке VisualBasic, которые разрабатываются пользователем для реализации нестандартных процедур обработки данных в задачах пользователя.

- **Страницы** показывают все ярлыки страниц доступа к данным в Internet. Дают возможность ввода, редактирования, просмотра и манипулирования данными из сети.

В каждой базе данных имеется **Окно базы данных MS Access**, которое состоит:

**Верхняя строка** – **Строка заголовка** окна БД, на которой находится имя файла БД, с которым вы работаете в данный момент.

Под строкой заголовка окна БД расположены три кнопки:

**Лента**- с вкладками: **Файл**, **Главная**, **Создание**, **Внешние данные**, **работа с БД**, **Работа с таблицами**.

**Рабочее поле** разделено на части: слева **Все объекты Access: Таблицы, Запросы, Формы, Отчеты**. Поле поиска. Справа **работа с объектами**.

The screenshot shows the Microsoft Access interface with a table named 'Учащиеся' (Students) open. The table has the following columns: No, Фамилия (Surname), Имя (Name), Дата (Date), пол (Gender), Улица (Street), Двор (Courtyard), Квартира (Apartment), Класс (Class), Группа (Group), Хобби (Hobbies), and Глаза (Eyes). The data is as follows:

No	Фамилия	Имя	Дата	пол	Улица	Двор	Квартира	Класс	Группа	Хобби	Глаза
1	Сухонов	Сергей	16.02.1981	Д	Чердынская	23	74	10 основная		тяжелая атлетика	зеленые
2	Перогов	Юрий	05.12.1983	Д	Нурбышева	6	31	8 основная		футбол	голубые
3	Лебедева	Света	16.06.1985	Ж	Пушкина	37	65	6 специальная		владение	карие
4	Голдобин	Сергей	23.05.1988	Д	Леонова	12	18	3 основная		ложи	голубые
5	Ельцина	Наталья	24.05.1982	Ж	Чердынская	37	46	9 специальная		чтение	серые
6	Сухонова	Наталья	20.12.1986	Ж	Ленина	12	22	3 подготовительная		шитье	зеленые
7	Петрова	Света	18.04.1982	Ж	Пушкина	37	3	9 основная		ложи	серые
8	Горина	Оля	20.12.1984	Ж	Свищева	66	99	7 подготовительная		аэробика	карие
9	Попов	Михаил	07.07.1987	Д	Леонова	72	6	4 подготовительная			голубые
10	Сергеев	Саша	30.11.1989	Д	Нурбышева	3	31	2 основная		каратэ	зеленые
11	Павлова	Елена	11.12.1985	Ж	Пушкина	5	6	5 основная		аэробика	карие
12	Емельянова	Наталья	25.05.1981	Ж	Попова	40	47	10 основная		шитье	зеленые
13	Евдокимов	Михаил	18.08.1980	Д	Чердынская	3	40	7 основная		футбол	зеленые
14	Басеева	Елена	14.10.1984	Ж	Ленина	14	82	9 основная		ложи	серые
15	Сухонова	Света	29.07.1980	Ж	Нурбышева	37	32	11 основная		аэробика	карие

Важным средством, облегчающим работу с **Access** для начинающих пользователей, являются **мастера** – это специальные программные надстройки, предназначенные для создания объектов базы данных в режиме последовательного диалога. Для опытных и продвинутых пользователей существуют возможности более гибкого управления ресурсами и возможностями в СУБД в режиме **конструктора**.

#### СОЗДАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ

Прежде чем приступить к созданию таблицы, необходимо спроектировать её структуру, которая определяется составом и последовательностью полей.

**1 способ:** Создание таблицы при помощи **Мастера**, который автоматически создает таблицу по одному из шаблонов, содержащих соответствующий набор полей.

**2 способ:** Создание таблицы в режиме **Конструктор**. При создании таблицы выводится пустая структура таблицы, в которую пользователь сам задаёт основные характеристики полей создаваемой таблицы: **имя поля, тип данных и свойства полей**.

#### ИМЯ ПОЛЯ

Каждое поле в таблице должно иметь уникальное имя, при задании которых придерживаются следующих правил:

1. Не должно содержать более 64 символов;

2. Может включать буквы, цифры, пробелы и специальные символы, за исключением точки (.), восклицательного знака (?) и прямых скобок | |;
3. Не может начинаться с пробела;
4. Должны быть уникальными.

#### ТИПЫ ДАННЫХ

Тип данных определяется значениями, которые предполагается вводить в поле, и операциями, которые будут выполняться с этими значениями. MS Access допускает использование десяти типов данных (табл.1). Для задания типа данных следует нажать кнопку раскрытия списка  , расположенную в правой части поля, и выбрать из списка требуемый тип.

Таблица 1

**Типы данных**

№	Тип данных	Размер	Содержимое
1	Текстовый	от 1 до 255 символов	Текст (по умолчанию — 50 символов)
2	Поле Мемо	от 1 до 65535 символов	Текст произвольной длины
3	Числовой	от 1 до 16 байт	Число
4	Дата/время	8 байт	Дата или время
5	Денежный	8 байт	Число, представляющее денежную сумму
6	Счетчик	4 байта	Число, автоматически увеличиваемое на единицу для каждой новой записи
7	Логический	1 бит	Логическое значение «True» или «False» («Истина» или «Ложь»)
8	Поле объекта OLE	до 1 Гбайта	OLE-объекты, созданные в других программах с использованием протокола OLE

#### СВОЙСТВА ПОЛЯ

Позволяют изменять способы хранения и отображения данных. Каждый тип данных наделен собственными свойствами. Например, если поле имеет тип данных Текстовый, то указывается его длина, если Числовой – формат поля процентный или экспоненциальный, число десятичных знаков и т.д.

#### СВЯЗЫВАНИЕ ТАБЛИЦ

Связи между таблицами устанавливаются в схеме данных. Схема данных является графическим образом базы данных и наглядно отображает таблицы и связи между ними, а также обеспечивает использование связей при обработке данных. В схеме данных устанавливаются параметры обеспечения целостности связей в базе данных.

При построении схемы данных Access автоматически определяет по выбранному полю связи тип отношения между таблицами.

Различают следующие типы связей:

- **Один-к-одному (1 — 1)** — создается в том случае, когда оба связующих поля являются ключевыми. При таком типе связи каждая запись одной таблицы соответствует только одной записи другой таблицы, и наоборот. Отношения этого типа используются не очень часто, поскольку большая часть сведений, связанных таким образом, может быть помещена в одну таблицу.

**Например**, даны две таблицы: **Студент** (номер, ФИО, Группа) и **Сессия** (Оценка1, Оценка2, Оценка3). Каждому номеру таблицы **Студент** соответствует один номер в таблице **Сессия**.

- **Один-ко-многим (1 — ∞)** — создается в том случае, когда одно из связующих полей является ключевым. Такое отношение означает, что каждая запись первой (главной) таблицы может быть связана с любым числом записей второй (подчиненной) таблицы, но каждая запись второй таблицы может быть связана не более чем с одной записью первой таблицы. Является наиболее часто используемым типом связи между таблицами.

**Например**, даны две таблицы: **Студент и Библиотека** (Код, Название книги, Автор, Год издания). Каждому номеру из таблицы **Студент** может соответствовать несколько кодов из таблицы **Библиотека**.

- **Многие-ко-многим (∞ — ∞)** — предполагает, что каждая запись одной таблицы может быть связана с любым числом записей другой таблицы, и наоборот.

**Например**, даны две таблицы: **Студент и Спортивные секции**. Каждому студенту может посещать несколько секций и каждую секцию могут посещать несколько студентов.

#### ПОНЯТИЕ ЗАПРОСА

**Запросы** позволяют выбрать данные из одной или нескольких взаимосвязанных таблиц БД, произвести над ними вычисления и получить результаты в виде таблиц.

Посредством запросов можно обновлять, удалять или добавлять данные в таблицы, а также создавать новые таблицы на основе уже существующих. Запросы строятся на основе одной или нескольких таблиц базы данных или на ранее созданных запросах.

#### ТИПЫ ЗАПРОСОВ

Запросы можно разделить на несколько типов: по способу формирования (проектирования) и по результатам действия.

**1. По способу формирования** различают следующие запросы:

- **Запросы по образцу** или **QBE - запросы**, для определения которых пользователь должен указать параметры запроса в окне конструктора, задавая образцы для поиска информации.

- **Структурированные запросы** или **SQL - запросы**, для определения которых пользователь должен описать запрос с помощью особого языка запросов, используя специальные команды и функции.

Начинающим пользователям рекомендуется использовать запросы по образцу. Они более наглядные, просты для изучения и понимания, при этом не требуется знание команд и функций языка структурированных запросов.

## 2. По результатам действия различают:

- **Запросы на выборку**— запросы этого типа возвращают данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений. Запрос на выборку является наиболее часто используемым типом запроса.

## 3. Запрос с параметрами

— это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные, например условие для возвращения записей или значение, которое требуется вставить в поле. Для того чтобы выводилось диалоговое окно для ввода конкретного значения поля, нужно определить параметр запроса. Имя параметра запроса вводится в строке условий отбора в квадратных скобках [...]. При выполнении запроса это имя появится в диалоговом окне **Введите значение параметра**.

- **Модифицирующие запросы (запрос на изменение)** — это запросы на выполнение действий, в результате выполнения которых изменяется содержимое таблиц базы данных, за одну операцию можно изменить или переместить несколько записей.

Существует четыре вида модифицирующих запросов:

- **Запросы удаления** позволяют удалить группу записей из одной или нескольких таблиц БД, удовлетворяющие определенным условиям.

- **Запросы добавления** используются для добавления группы записей из одной или нескольких таблиц в конец одной или нескольких таблиц, удовлетворяющих определенным условиям.

- **Запросы обновления** позволяют вносить общие изменения в группу записей одной или нескольких таблиц, удовлетворяющих определенным условиям.

- **Запросы создания таблиц** позволяют создать новую таблицу на основе всех или части данных из одной или нескольких таблиц в базе данных.

- **Перекрестные запросы** — используют для расчетов и представления данных в структуре (матрицы), облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.

- **Многотабличные запросы**— позволяют сформировать новые таблицы, записи которых образуются путем объединения взаимосвязанных записей из разных таблиц БД и включения нужных полей из этих таблиц.

• **Итоговые запросы** — выполняющие вычисления над группой записей. Для выполнения итоговых вычислений следует дать команду **Вид→Групповые операции** или нажать кнопку  $\Sigma$  на Панели инструментов.

#### Типы итоговых операций

Значение	Операция
Sum	Сложение
Avg	Среднее значение
Min	Минимальное значение
Max	Максимальное значение
Count	Количество записей, вошедших в группу

#### ЭКРАННЫЕ ФОРМЫ

В MS Access существует два способа ввода данных в БД:

1. Непосредственно в таблицы базы данных;
2. С помощью специально разработанных экранных форм.

Первый способ используют в том случае, если данные изменяются достаточно редко, а база данных имеет простую структуру, т.е. в ней нет сложных связей между таблицами. Второй способ, если приходится часто изменять данные, а база данных состоит из большого количества таблиц, связанных друг с другом.

Формы позволяют вычислять значения и выводить на экран результат. Источником данных для формы являются записи таблицы или запроса.

Многотабличная (составная) форма обычно применяется для отображения информации из нескольких взаимосвязанных таблиц. Она состоит из основной части (**главной формы**) и одной или нескольких **подчиненных форм**. Подчиненные формы используются для просмотра и редактирования данных из таблиц с типом связи **один ко многим**.

Форма может быть создана с помощью конструктора или мастера форм.

#### ПОНЯТИЕ ОТЧЕТА

В отчете можно наглядно представить извлеченную из базы данных информацию, дополнив ее результатами анализа и вычислений. Отчет может быть создан на основе данных одной или нескольких взаимосвязанных таблиц или запросов.

Средства MS Access позволяют создавать отчеты как в табличном виде, так и в свободной форме. При этом могут быть выполнены самые высокие требования к оформлению документов.

Основными объектами отчета являются поля отчета и пояснительный текст. Для улучшения восприятия информации в отчет могут добавляться OLE объекты, рисунки, линии и прямоугольники. Поля отчета могут непосредственно соответствовать полям исходной таблицы (запроса) или являться результатом вычислений над ними.

В MS Access для создания отчетов используются: **Конструктор отчетов, Мастер отчетов, Мастер диаграмм** и др.

Мастер позволяет создавать отчеты с группировкой записей и представляет собой простейший способ создания отчетов. Он помещает выбранные поля в отчет и предлагает шесть стилей его оформления. После завершения работы Мастера полученный отчет можно доработать в режиме Конструктора.

Формы и отчеты данных включают следующие области: заголовок формы (отчета), верхний колонтитул, область данных, нижний колонтитул и примечание формы (отчета).

Заголовок формы (отчета)
Верхний колонтитул
Область данных
Нижний колонтитул
Примечание формы (отчета)

### 6.3. Вопросы для самопроверки по теме 6

*Задание №1.* База данных, представленная в виде таблицы, называется:

1. иерархической;
2. сетевой;
3. реляционной;
4. пространственной.

*Задание №2.* Объекты СУБД MS Access, позволяющие выбирать из базы данных только необходимую информацию, называются:

1. таблицы;
2. запросы;
3. отчеты;
4. формы.

*Задание №3.* Столбец таблицы базы данных, содержащий значения определенного свойства, это:

1. запись;
2. ключ;
3. запрос;
4. поле.

*Задание №4.* Объектами СУБД не являются:

1. запросы;
2. окна;
3. отчеты;
4. макросы.

*Задание №5.* База данных - это ...

1. совокупность данных, организованных по определенным правилам;
2. совокупность программ для хранения и обработки больших массивов информации;
3. интерфейс, поддерживающий наполнение и манипулирование данными;
4. определенная совокупность информации.

## **7. ОСНОВЫ ЗАЩИТЫ ИНФОРМАЦИИ**

### **7.1. Защита информации как закономерность развития компьютерных систем**

*Защита информации* – это применение различных средств и методов, использование мер и осуществление мероприятий для того, чтобы обеспечить систему надежности передаваемой, хранимой и обрабатываемой информации.

Защита информации включает в себя:

- обеспечение физической целостности информации, исключение искажений или уничтожения элементов информации;
- недопущение подмены элементов информации при сохранении ее целостности;
- отказ в несанкционированном доступе к информации лицам или процессам, которые не имеют на это соответствующих полномочий;
- приобретение уверенности в том, что передаваемые владельцем информационные ресурсы будут применяться только в соответствии с обговоренными сторонами условиями.

Процессы по нарушению надежности информации подразделяют на случайные и злоумышленные (преднамеренные). Источниками *случайных* разрушительных процессов являются непреднамеренные, ошибочные действия людей, технические сбои. *Злоумышленные* нарушения появляются в результате умышленных действий людей.

Проблема защиты информации в системах электронной обработки данных возникла практически одновременно с их созданием. Ее вызвали конкретные факты злоумышленных действий над информацией.

Если в первые десятилетия активного использования ПК основную опасность представляли хакеры, подключившиеся к компьютерам в основном через телефонную сеть, то в последнее десятилетие нарушение надежности информации прогрессирует через программы, компьютерные вирусы, глобальную сеть Интернет.

Имеется достаточно много способов несанкционированного доступа к информации, в том числе:

- просмотр;
- копирование и подмена данных;

- ввод ложных программ и сообщений в результате подключения к каналам связи;
- чтение остатков информации на ее носителях;
- прием сигналов электромагнитного излучения и волнового характера;
- использование специальных программ.

### **Объекты и элементы защиты в компьютерных системах обработки данных**

*Объект защиты* – это такой компонент системы, в котором находится защищаемая информация. *Элементом защиты* является совокупность данных, которая может содержать необходимые защите сведения.

При деятельности компьютерных систем могут возникать:

- отказы и сбои аппаратуры;
- системные и системотехнические ошибки;
- программные ошибки;
- ошибки человека при работе с компьютером.

Для обеспечения безопасности проводятся разные мероприятия, которые объединены понятием «система защиты информации».

*Система защиты информации* – это совокупность организационных (административных) и технологических мер, программно-технических средств, правовых и морально-этических норм, которые применяются для предотвращения угрозы нарушителей с целью сведения до минимума возможного ущерба пользователям и владельцам системы.

*Организационно-административными средствами защиты* называется регламентация доступа к информационным и вычислительным ресурсам, а также функциональным процессам систем обработки данных. Эти средства защиты применяются для затруднения или исключения возможности реализации угроз безопасности. Наиболее типичными организационно-административными средствами являются:

- допуск к обработке и передаче охраняемой информации только проверенных должностных лиц;
- хранение носителей информации, которые представляют определенную тайну, а также регистрационных журналов в сейфах, недоступных для посторонних лиц;
- учет применения и уничтожения документов (носителей) с охраняемой информацией;
- разделение доступа к информационным и вычислительным ресурсам должностных лиц в соответствии с их функциональными обязанностями.

*Технические средства защиты* применяются для создания некоторой физически замкнутой среды вокруг объекта и элементов защиты. При этом используются такие мероприятия, как:

- ограничение электромагнитного излучения через экранирование помещений, в которых осуществляется обработка информации;

- реализация электропитания оборудования, обрабатывающего ценную информацию, от автономного источника питания или общей электросети через специальные сетевые фильтры.

*Программные средства и методы защиты* являются более активными, чем другие применяемые для защиты информации в ПК и компьютерных сетях. Они реализуют такие функции защиты, как разграничение и контроль доступа к ресурсам; регистрация и изучение протекающих процессов; предотвращение возможных разрушительных воздействий на ресурсы; криптографическая защита информации.

Под *технологическими средствами защиты информации* понимаются ряд мероприятий, органично встраиваемых в технологические процессы преобразования данных. В них также входят:

- создание архивных копий носителей;
- ручное или автоматическое сохранение обрабатываемых файлов во внешней памяти компьютера;
- автоматическая регистрация доступа пользователей к различным ресурсам;
- выработка специальных инструкций по выполнению всех технологических процедур и др.

*Правовые и морально-этические меры и средства защиты* включают в себя действующие в стране законы, нормативные акты, регламентирующие правила, нормы поведения, соблюдение которых способствует защите информации.

## **7.2. Средства опознавания и разграничения доступа к информации**

Пароль как средство обеспечения безопасности способен использоваться для идентификации и установления подлинности терминала, с которого входит в систему пользователь, а также для обратного установления подлинности компьютера по отношению к пользователю.

### **Криптографический метод защиты информации**

**Криптография (шифрование)** – это кодирование данных, посылаемых в сеть, так, чтобы их могли прочитать только стороны, участвующие в конкретной операции. Надёжность защиты информации зависит от алгоритма шифрования и длины ключа в битах.

**Метод шифрования** – это алгоритм, описывающий порядок преобразования исходного сообщения в результирующее.

**Ключ шифрования** – это набор параметров, необходимых для применения метода шифрования. При компьютерном шифровании ключ представляется как последовательность символов, сохранённых на жёстком или съёмном диске. Различают *статические* варианты ключей – они не меняются при работе с разными сообщениями и *динамические ключи* – они изменяются для каждого сообщения.

### *Типы методов шифрования:*

- симметричные: один и тот же ключ используется и для шифровки, и для дешифровки. Такие методы неудобны в электронной коммерции, так как у продавца и покупателя должны быть разные права к доступу информации. Продавец посылает всем покупателям одни и те же каталоги, но покупатели возвращают ему конфиденциальную информацию о своих кредитных картах, и нельзя смешивать заказы и их оплату для разных покупателей;
- асимметричные (несимметричные): основываются на специальных математических методах, которые создают пару ключей так, что то, что зашифровано одним ключом, может быть дешифровано только другим, и наоборот. Один из ключей называется открытым, его может получить каждый желающий. Второй ключ разработчик ключа оставляет себе, он называется закрытым (секретным). Например, в электронной коммерции продавец создает пару ключей. Закрытый ключ он оставляет себе, открытый посылает покупателю. Если клиент получил файл, к которому не подходит его ключ, значит, его послала не та фирма, с которой он ведет деловую переписку. Покупатели шифруют свои заказы, договоры своим открытым ключом и посылают их продавцу. Их дешифровку может сделать только владелец закрытого ключа, т. е. продавец.

**Защищённый канал** – способ передачи сообщений, при котором обе стороны используют один и тот же метод шифрования, известный только им.

**Электронно-цифровая подпись** – код специальной структуры, который позволяет однозначно связать содержание документа, пересылаемого по сети, и его автора (аутентифицировать документ).

**Хэш-функции** – это функции, которые позволяют из одной последовательности чисел получить другую последовательность таким образом, что обратное преобразование невозможно.

**Хэширование** – обработка некоего сообщения хэш-функцией, при которой двоичные коды сообщения воспринимаются как коды двоичных чисел. В результате создается уникальная последовательность символов фиксированной длины, которая однозначно соответствует содержанию исходного сообщения. Эту последовательность обозначают термином хэш-код. Смысл хеширования можно проиллюстрировать бытовым примером: для того, чтобы не забыть какой-нибудь тюк багажа в пути, мы запоминаем количество мест багажа, и, чтобы проверить, не забыли ли мы что-то, просто пересчитываем тюки. Но, зная только количество тюков, мы не сможем получить информацию о том, что в них содержалось.

**Дайджест сообщения, электронная печать, сводка сообщения** – уникальная последовательность символов фиксированной длины, которая создаётся на основе содержания электронного документа. Дайджест получается путём хеширования. Он уникален для сообщения, как отпечатки пальцев для человека. Если изменить хотя бы один символ в документе, хэш-код станет другим.

### 7.3. Компьютерные вирусы

*Компьютерный вирус* – это специально написанная программа, способная самопроизвольно присоединяться к другим программам (заражать их), создавать свои копии и внедрять их в файлы, системные области компьютера и другие объединенные с ним компьютеры в целях нарушения нормальной работы программ, порчи файлов и каталогов, а также создания разных помех при работе на компьютере.

Появление вирусов в компьютере определяется по следующим наблюдаемым признакам:

- уменьшение производительности работы компьютера;
- невозможность и замедление загрузки ОС;
- повышение числа файлов на диске;
- замена размеров файлов;
- периодическое появление на экране монитора неуместных сообщений;
- уменьшение объема свободной ОП;
- резкое возрастание времени доступа к жесткому диску;
- разрушение файловой структуры;
- загорание сигнальной лампочки дисковода, когда к нему нет обращения.

Каналы распространения вирусов:

- электронная почта;
- интернет-сайты;
- рассылки графических и аудиофайлов;
- непроверенные съёмные диски.

Классификация вирусов:

#### **Классификация вирусов**

Множество компьютерных вирусов, существующих на современном этапе можно разделить на несколько групп.

#### ***По среде обитания***

1. Сетевые
2. Файловые
3. Загрузочные
4. Файлово-загрузочные

*Сетевые* вирусы распространяются по различным сетям, т.е. при передаче информации с одного компьютера на другой, соединенные между собой сетью, например Интернет.

*Файловые* вирусы заражают исполнительные файлы и загружаются после запуска той программы, в которой он находится. Файловые вирусы могут внедряться и в другие файлы, но записанные в таких файлах, они не получают управление и теряют способность к размножению.

*Загрузочные* вирусы внедряются в загрузочный сектор дискет или логических дисков, содержащий программу загрузки.

*Файлово-загрузочные* вирусы заражают одновременно файлы и загрузочные сектора диска.

***По способу заражения среды обитания***

1. Резидентные
2. Нерезидентные

*Резидентный вирус* при заражении компьютера оставляет в оперативной памяти свою резидентную часть, которая потом перехватывает обращение операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т.п.) и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера.

*Нерезидентный вирус* не заражает память компьютера и является активным ограниченное время. Активизируются в определенные моменты, например при обработке документов текстовым процессором.

***По деструктивным (разрушительным) возможностям***

1. Безвредные
2. Неопасные
3. Опасные
4. Очень опасные

*Безвредные* вирусы проявляются только в том, что уменьшают объем памяти на диске в результате своего распространения.

*Неопасные*, так же уменьшают объем памяти, не мешают работе компьютера, такие вирусы порождают графические, звуковые и другие эффекты.

*Опасные вирусы*, которые могут привести к различным нарушениям в работе компьютера, например к зависанию или неправильной печати документа.

*Очень опасные*, действие которых может привести к потере программ, данных, стиранию информации в системных областях памяти и даже приводить к выходу из строя движущихся частей жесткого диска при вводе в резонанс.

***По особенностям алгоритма***

1. Паразитические
2. Репликаторы
3. Невидимки
4. Спутники
5. Студенческие
6. Мутанты
7. Троянские

*Паразитические* – это одни из самых простых вирусов. Они изменяют содержимое файлов и секторов диска и могут быть достаточно легко обнаружены и уничтожены.

*Вирусы-репликаторы* (черви) распространяются по компьютерным сетям, вычисляют адреса сетевых компьютеров и записывают по этим адресам свои копии.

*Вирусы невидимки* (стелс-вирусы) – вирусы, которые очень трудно обнаружить и обезвредить, так как они перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо своего незараженные участки диска.

*Мутанты*(призраки)содержат алгоритмы шифровки-расшифровки, благодаря которым копии одного и того же вируса не имеют ни одной повторяющейся цепочки байтов. Такие вирусы самые сложные в обнаружении.

*Троянские программы* (квазивирусы) не способны к самораспространению, но очень опасны, так как, маскируясь под полезную программу, разрушают загрузочный сектор и файловую систему дисков.

*Спутники* – вирус, который не изменяет файл, а для выполнимых программ (exe) создают одноименные программы типа com, которые при выполнении исходной программы запускаются первыми, а затем передают управление исходной выполняемой программе.

*Студенческие вирусы* представляют собой самые простые и легко обнаруживаемые вирусы.

Однако четкого деления между ними не существует, и все они могут составлять комбинацию вариантов взаимодействия - своеобразный вирусный "коктейль".

#### **7.4. Антивирусные программы**

Широкое распространение компьютерных вирусов привело к разработке антивирусных программ, которые позволяют обнаруживать и уничтожать вирусы, «лечить» пораженные ресурсы.

Основой работы большинства антивирусных программ является принцип поиска сигнатуры вирусов. *Вирусной сигнатурой* называют некоторую уникальную характеристику вирусной программы, выдающую присутствие вируса в компьютерной системе. Чаще всего в антивирусные программы включается периодически обновляемая база данных сигнатур вирусов. Антивирусная программа изучает и анализирует компьютерную систему, а также проводит сравнение, отыскивая соответствие с сигнатурами в базе данных. Если программа находит соответствие, она старается вычистить обнаруженный вирус.

##### **Классификация антивирусных программ**

В настоящее время существует множество программ, выполняющих антивирусную защиту. Их можно классифицировать по следующим признакам:

##### ***По принципу работы***

– *мониторы* – являются частью ОС, автоматически проверяют на вирусы все поступающие файлы, все открываемые и закрываемые файлы,

если нужно – попутно лечат. К программам такого типа относятся NOD32, AVP (Antiviral Toolkit Pro – лаборатория Касперского);

– *сканеры*– проверяют файлы, находящиеся в ОЗУ, выдают сообщения о подозрительных файлах, но не лечат их.

– *брандмауэр (firewall)* – программа, которая является фильтром при обмене информацией между внешней и локальной сетью. Она разрешает доступ к информации сети только уполномоченным лицам и задерживает файлы тех типов, которые указаны администратором сети (вирусы, баннеры, файлы запрещенной тематики).

#### ***По принципу обнаружения вирусов:***

– *полифаги*: просматривают коды программ на наличие в них известных вирусных фрагментов или фрагментов кода, типичных для вирусных действий, если можно – лечат. Качество работы сильно зависит от даты обновления вирусной базы. Наиболее популярные программы этого типа: NOD32, AVP, Doctor Web, Norton Antivirus,

– *ревизоры*: только диагностируют, заражён ли файл. Для этого они сверяют эталонные контрольные суммы всех неизменяемых файлов в компьютере с контрольными суммами на момент проверки. Те файлы, у которых они не совпали – заражены. Качество работы зависит от того, был ли заражён компьютер в момент создания базы эталонных контрольных сумм при загрузке ревизора на компьютер. Примером программы такого типа является Aids test,

– *эвристические анализаторы*: реагируют на фрагменты кода, которые выполняют действия, похожие на вирусные, но не зарегистрированы в антивирусных базах.

Другие типы антивирусных программ: ***фаги, дезинфекторы, вирус-фильтры, иммунизаторы, детекторы, вакцины.***

Помимо применения антивирусных программ для уменьшения вероятности потери и порчи информации из-за вирусов рекомендуется периодически архивировать важную информацию и хранить ее на съемных носителях. Это позволяет при обнаружении испорченных вирусом файлов восстановить их предыдущие версии, а не создавать их заново. Основные архиваторы, которые используются в настоящее время, это: 7-Zip, ARJ, WinRAR, WinZip. Основные расширения архивов: \*.ZIP, \*.RAR, \*.ARJ.

## **7.5. Облачная антивирусная защита**

Активное развитие телекоммуникационных систем в настоящее время сделало возможным предоставление конечному пользователю доступа к услугам, приложениям и вычислительным ресурсам различной мощности, находящимся на разных серверах. Такие виды услуг называются интернет-облаками или облачными вычислениями.

Одна из услуг, относящаяся к облачным вычислениям, – это антивирусные облака. Производители антивирусных программ имеют в своем распоряжении гораздо более мощное оборудование и программное

обеспечение для распознавания вирусных сигнатур (участков кода, которые могут нанести вред ПО компьютера, файлов в критических местах ЭВМ), чем то, которое имеет конечный пользователь. При работе в облаке антивирусные программы на компьютерах пользователей все подозрительные файлы, которые они не могут однозначно определить как вирусы, отправляют на сервер своей фирмы-производителя для анализа в экспертной системе. Сервер производителя, со своей стороны, выставляет по выполняемым операциям баллы опасности присланного приложения и расширяет, если нужно, базу вирусных сигнатур. Так как на сервер стекается информация от многих пользователей, такая технология ускоряет обнаружение новых вариантов вирусов и время реакции на них, а также обеспечивает обновление антивирусных баз всех своих пользователей частыми и маленькими порциями.

## **7.6. Вопросы для самопроверки по теме 7**

*Задание №1.* Результатом реализации угроз информационной безопасности может быть

1. уничтожение устройств ввода/вывода;
2. блокирование информации;
3. несанкционированный доступ к информации;
4. уничтожение информации;
5. компьютерное мошенничество;
6. внедрение дезинформации.

*Задание №2.* Несанкционированный доступ к информации – это:

1. незапланированный доступ к информации;
2. доступ к информации, нарушающий установленные правила разграничения доступа;
3. доступ к информации человека, не обладающего нужной квалификацией;
4. открытый доступ к информации.

*Задание №3.* Заражение компьютерными вирусами может произойти в процессе:

1. печати на принтере;
2. работы с файлами;
3. форматирования дискеты;
4. выключения компьютера.

*Задание №4.* Электронная цифровая подпись служит:

1. для защиты от незаконного копирования документа;
2. для доказательства ценности документа;
3. для установления целостности документа;
4. для защиты документа от подделки;

5. для определения объема документа;
6. для предотвращения отказа от авторства.

*Задание №5.* Укажите типы вирусных программ.

*Задание № 6.* Укажите классификацию антивирусных программ по принципу работы.

## 8. ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

### 8.1. Классификация вычислительных сетей

**Компьютерная сеть** – это два или больше компьютеров, связанных каналами передачи информации. Цель создания сетей – обеспечение совместного доступа к **сетевым ресурсам**.

*Виды сетей по географическому признаку и размерам:*

– **локальная:** сеть предприятия или учреждения, в которой рабочие станции распределены на расстоянии не более 300-500м. Обозначается аббревиатурами ЛВС или LAN (Local Areal Net);

– **региональные и корпоративные:** объединяют компьютеры большого региона или филиалы учреждений, разбросанные в пределах города, большого географического региона, страны. Обозначается аббревиатурой MAN. (Metropolitan Areal Net);

– **глобальные:** объединяют всех абонентов (LAN и MAN) вне зависимости от места их расположения: страны, континента, всей земли, например, Интернет. Обозначается аббревиатурой WAN. (World Areal Net).

*Виды сетей по возможностям передачи сигналов:*

– **узкополосные:** канал связи может передавать только один сигнал в любой момент времени (телефонная линия);

– **широкополосные:** одновременно можно передавать несколько сигналов, используя для каждого свою частоту передачи (кабельное телевидение).

### 8.2. Виды сетевых ресурсов

– **аппаратные:** общий принтер, общий жёсткий диск для хранения программ и данных отдельных пользователей;

– **программные:** для выполнения сложных и длительных расчётов можно подключиться к мощной ЭВМ, послать на неё задание на расчёты и исходные данные, и, по окончании расчётов просто получить готовые результаты.

– **информационные:** всевозможные справочные данные, архивы научных работ, книг и т. п.

*Виды каналов передачи информации:*

– **специальные кабели:** коаксиальный кабель, оптоволоконный кабель (самая надёжная и быстрая связь), витая пара – телефонный кабель;

– **электромагнитные волны разных частот:** спутниковая связь, подключение ноутбуков, мобильных телефонов;

Важнейшая характеристика канала – **скорость передачи информации.** В ЛВС обычно от 10 до 100 Мбит/сек., в крупных сетях –

На скорость передачи информации влияет также тип ее кодировки: чем короче кодировка символов, тем меньше бит она содержит и, следовательно, тем быстрее она передается по тому же самому каналу.

### 8.3. Топология и архитектура вычислительных сетей

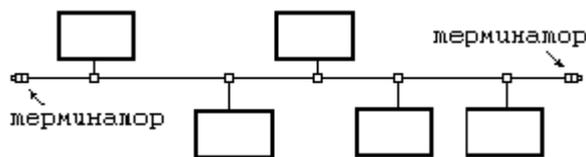
**Топология сети** – это логический и физический способ соединения компьютеров.

Различают следующие базовые варианты топологий:

– **одноранговая:** все компьютеры имеют одинаковые права (обычно до 10 рабочих станций);

– **сети на основе сервера:** сервер – это компьютер, на котором хранится информация, необходимая разным пользователям или же выполняющий работы по поддержанию связи между компьютерами. Другое значение этого термина – программа, выполняющая запросы с рабочих станций на доступ к ресурсам. **Рабочая станция (клиент)** – это внутренний компьютер ЛВС. Другое значение этого термина – программа, создающая запрос серверу на какие-либо ресурсы;

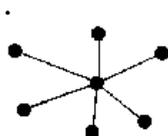
– **линейная (шинная) топология:** соединяющий кабель последовательно проходит от одного компьютера к другому:



– **кольцевая топология:**



– **звездообразная топология:**



**Архитектура сети** – это понятие, которое включает в себя топологию сети, состав ее устройств, правила их взаимодействия, кодирование, адресацию и передачу информации, управление потоком сообщений, контроль ошибок, анализ работы в аварийных ситуациях.

### **Основные виды архитектур (физический уровень):**

- *Ethernet*: широковещательная сеть, т.е. все рабочие станции могут принимать все сообщения. Топология линейная или звездообразная.
- *Token Ring*: топология кольцевая. Каждый узел ожидает своей очереди на посылку сообщения.
- *FDDI*: высокоскоростная передача данных по оптоволоконным линиям. Топология смешанная (кольцевая + древовидная)
- *ATM*: передача цифровых данных, видеоинформации и голоса по одним и тем же линиям.
- *Wi-Fi, IrDa*: беспроводная (радиорелейная) связь.

### **8.4. Программное обеспечение вычислительных сетей**

*Сетевые протоколы* – это наборы правил для обмена информацией в сети и разработки сетевого оборудования. Более подробно их можно определить как стандарты, которые определяют формы представления и способы пересылки разных типов сообщений между компьютерами, а также правила совместной работы различного оборудования в сетях, необходимые при создании этого оборудования.

Совокупность протоколов, которые используют компьютеры при работе в сети, обозначается термином *стек*. Традиционно стек делится на 7 уровней, функции которых определяются эталонной *моделью взаимодействия открытых систем* (сетевая модель OSI – Open System Interconnection):

– *Физический уровень (1)*: процедуры управления аппаратурой передачи данных и подключённым к ней каналам связи. Примеры протоколов: Ethernet, Arc net, Token Ring.

– *Канальный уровень (2)*: отвечает за передачу информации по логическому каналу, установленному между двумя ЭВМ, соединёнными физическим каналом. На этом уровне обнаруживаются ошибки передачи пакетов, реализуется алгоритм восстановления информации в случае обнаружения сбоев или потери данных. Детализируются способы осуществления непосредственной связи объектов сетевого уровня. Примеры протоколов: LAP-B, SNAP, HDLC.

– *Сетевой уровень (3)*: отвечает за маршрутизацию пакетов и связь между разными сетями, т.е. устанавливает логические каналы между объектами для реализации протоколов транспортного уровня. Примеры протоколов: IP (Internet Protocol), IPX, IDN, X.25.

– *Транспортный уровень (4)*: определяет интерфейс между процессами и сетью, т.е. устанавливает логические каналы между процессами и обеспечивает передачу информационных пакетов, сформированных по правилам сетевого уровня. Примеры протоколов: TCP (Transmission Control Protocol), UDP, NSP, X.224, NetBIOS.

– *Сеансовый уровень (5)*: осуществляет установку и поддержку сеанса связи между двумя абонентами. Определяются средства, необходимые для

синхронизации и управления обменом данными между сетевыми объектами. Примеры протоколов: RPC, X.225.

– **Представительский уровень (6):** определяет синтаксис данных, т.е. коды и форматы данных, посылаемых в сеть. Каждая прикладная программа, каждая операционная система имеет свой способ кодировки данных и команд. На этом уровне устанавливаются стандартные способы кодировки для информации, которая выходит из локальной сети. Аппаратное обеспечение, обеспечивающее перекодировку информации при выходе из локальной сети, обозначается термином шлюз. Примеры протоколов: X.226

– **Прикладной уровень (7):** совокупность правил для разработки программ-приложений, которые делают запросы к ресурсам, расположенным в сети. В него входит регламентация всех работ, которые связаны с запуском программ пользователя и их выполнением. Примеры протоколов этого уровня: HTTP, SMTP, FTP, DNS, POP3. В рамках этих протоколов пользователь делает запросы на использование сетевых ресурсов.

**Основной протокол**, под управлением которого работает **Интернет**, – это **TCP/IP**. Другими словами, совокупность протоколов TCP/IP является основой построения и функционирования сети Интернет.

## 8.5. Протоколы электронной почты

Серверная часть пакета программ, обслуживающих электронную почту, состоит из трёх основных подсистем: подсистемы хранения сообщений, транспортной подсистемы, службы каталогов.

**Подсистема хранения** обеспечивает хранение и разделение пришедшей почты с помощью учётных записей. Учётная запись содержит учётное имя пользователя (логин: название почтового ящика) и ряд другой информации.

**Транспортная подсистема** обеспечивает пересылку исходящих сообщений от своих клиентов и транзитных сообщений.

**Служба каталогов** обеспечивает хранение и корректировку учётных записей, направление сообщения именно тому, кому оно направлено.

Основные протоколы:

**SMTP (Simple Mail Transfer Protocol)** – простой протокол передачи исходящих сообщений. Он регламентирует передачу только символьной информации;

**POP (Post Office Protocol)** – используется для работы с входящей почтой

**MIME (Multipurpose Internet Mail Extension)** – многоцелевое расширение почты Интернета. Этот протокол регламентирует пересылку писем с присоединёнными файлами музыки или изображений.

## 8.6. Коммуникационное оборудование

**Повторитель (репитер)** – передаёт электрические сигналы от одного участка кабеля к другому, предварительно усиливая их и восстанавливая их форму. Используется в локальных сетях для увеличения их протяжённости. В терминологии OSI функционирует на физическом уровне.

**Коммутаторы** – многопортовые повторители, которые считывают адрес назначения каждого входящего пакета и передают его только через тот порт, который соединён с компьютером получателем. Могут функционировать на разных уровнях OSI.

**Концентратор (hub)** – многопортовое устройство для усиления сигналов при передаче данных. Используется для добавления в сеть рабочих станций или для увеличения расстояния между сервером и рабочей станцией. Работает как коммутатор, но вдобавок может усиливать сигнал.

**Мультиплексор (устройство или программа)** – позволяет передавать по одной коммуникационной линии одновременно несколько различных сигналов.

**Шлюз** – подготавливает данные к передаче между сетями или прикладными программами, использующими разные протоколы (способы кодировки, физические среды для передачи данных). Например, при подключении локальной сети к глобальной. Функционирует на прикладном уровне.

**Мост** – соединяет две сети с одинаковыми протоколами, усиливает сигнал и пропускает только те сигналы, которые адресованы компьютеру, находящемуся по другую сторону моста. Мостом также называют компьютер с двумя сетевыми картами, предназначенный для соединения сетей.

**Маршрутизатор** – соединяет разные ЛВС, как и мост, пропускает только ту информацию, которая предназначена для сегмента, с которым он соединён. Отвечает за выбор маршрута передачи пакетов между узлами. Выбор маршрута осуществляется на основе протокола маршрутизации, содержащего информацию о топологии сети, и специального алгоритма маршрутизации. Функционирует на сетевом уровне OSI.

## 8.7. Сеть Интернет

Сеть **Интернет** – это наиболее популярная глобальная сеть, объединяющая в себе многие глобальные, региональные и локальные сети. Название сети так буквально и переводится – «межсетевая

**Интернет** — гигантская всемирная компьютерная сеть, объединяющая десятки тысяч сетей всего мира. Её назначение — обеспечить любому желающему постоянный доступ к любой информации. Интернет предлагает практически неограниченные информационные ресурсы, полезные сведения, учёбу, развлечения, возможность общения с

компетентными людьми, услуги удалённого доступа, передачи файлов, электронной почты и многое другое. Интернет обеспечивает принципиально новый способ общения людей, не имеющий аналогов в мире.

Сеть была создана в 1984 году, и сейчас ею пользуются примерно сорок миллионов человек. Интернет всё время изменяется, поскольку имеет много квалифицированных пользователей, которые пишут программы для себя, а затем распространяют их среди желающих.

Самый распространённый и недорогой способ — посредством модема и телефонной линии. При этом используются три типа подключения, отличающиеся друг от друга по объёму услуг и цене:

1. **почтовое** — позволяет только обмениваться электронной почтой с любым пользователем Интернет, самое дешёвое;
2. **сеансное** в режиме *on-line* ("на прямом проводе") — работа в диалоговом режиме — все возможности сети на время сеанса;
3. **прямое** (личное), самое дорогостоящее — все возможности в любое время.

При работе в сеансном режиме доступ к Интернет обычно покупается у **провайдеров** (англ. provide — предоставлять, обеспечивать) — фирм, предоставляющих доступ к некоторой части Интернет и поставляющих её пользователям разнообразные услуги.

Отдельные участки Интернет представляют собой сети различной архитектуры, которые связываются между собой с помощью **маршрутизаторов**.

Передаваемые данные разбиваются на небольшие порции, называемые **пакетами**. Каждый пакет перемещается по сети независимо от других пакетов.

Сети в Интернет **неограниченно коммутируются** (т.е. связываются) **друг с другом**, потому что все компьютеры, участвующие в передаче данных, используют единый протокол коммуникации **ТСР/ІР** (читается "ти-си-пи / ай-пи"). На самом деле протокол ТСР/ІР — это два разных протокола, определяющих различные аспекты передачи данных в сети:

- **протокол ТСР** (Transmission Control Protocol) — протокол управления передачей данных, использующий автоматическую повторную передачу пакетов, содержащих ошибки; этот протокол отвечает за разбиение передаваемой информации на пакеты и правильное восстановление информации из пакетов получателя;
- **протокол ІР** (Internet Protocol) — протокол межсетевого взаимодействия, отвечающий за адресацию и позволяющий пакету на пути к конечному пункту назначения проходить по нескольким сетям.

**Протоколы** — это специальные стандарты, обеспечивающие совместимость компонентов компьютерных сетей.

**Сетевые протоколы** — стандарты, определяющие формы представления и способы пересылки сообщений, процедуры их

интерпретации, правила совместной работы различного оборудования в сетях.

Каждый компьютер, подключенный к сети Интернет имеет два равноценных уникальных адреса: **цифровой IP-адрес и символический доменный адрес**. Присваивание адресов происходит по следующей схеме: международная организация Сетевой информационный центр выдает группы адресов владельцам локальных сетей, а последние распределяют конкретные адреса по своему усмотрению.

IP-адрес компьютера имеет длину 4 байта. Обычно первый и второй байты определяют **адрес сети**, третий байт определяет **адрес подсети**, а четвертый — **адрес компьютера в подсети**. Для удобства IP-адрес записывают в виде четырех чисел со значениями от 0 до 255, разделенных точками, например: 145.37.5.150. Адрес сети — 145.37; адрес подсети — 5; адрес компьютера в подсети — 150.

**Доменный адрес** (англ. domain — область), в отличие от цифрового, является символическим и легче запоминается человеком. Пример доменного адреса: **barsuk.les.nora.ru**. Здесь домен **barsuk** — имя реального компьютера, обладающего IP-адресом, домен **les** — имя группы, присвоившей имя этому компьютеру, домен **nora** — имя более крупной группы, присвоившей имя домену **les**, и т.д. В процессе передачи данных доменный адрес преобразуются в IP-адрес.

## 8.8. Основные возможности, предоставляемые сетью Интернет

### 1. *World Wide Web* — главный информационный сервис.

World Wide Web (WWW, "Всемирная паутина") — гипертекстовая, а точнее, гипермедийная информационная система поиска ресурсов Интернет и доступа к ним.

Всё содержимое системы WWW состоит из **WWW-страниц**.

**WWW-страницы** — гипермедийные документы системы World Wide Web. Создаются с помощью языка разметки гипертекста HTML (Hypertext markup language).

При работе с системой WWW пользователи имеют дело с **программами-клиентами системы, называемыми браузерами**.

**Браузеры** (англ. Browse — листать, просматривать) — программы, с помощью которых пользователь организует диалог с системой WWW: просматривает WWW страницы, взаимодействует с WWW-серверами и другими ресурсами в Интернет.

Существуют сотни программ-браузеров. Самые популярные браузеры: **NetscapeNavigator** и **MicrosoftInternetExplorer**.

### 2. *Электронная почта*

Электронная почта (Electronicmail, англ. mail — почта, сокр. e-mail) служит для передачи текстовых сообщений в пределах Интернет, а также между другими сетями электронной почты. К тексту письма современные почтовые программы позволяют прикреплять звуковые и графические файлы, а также двоичные файлы — программы. При использовании

электронной почты каждому абоненту присваивается уникальный почтовый адрес, формат которого имеет вид:

<имя пользователя> @ < имя почтового сервера>.

Например: earth@space.com, где earth — имя пользователя, space.com — имя компьютера, @ — разделительный символ "эт коммерческое".

### 3. Система телеконференций Usenet (от Users Network).

Эта система организует коллективные обсуждения по различным направлениям, называемые телеконференциями. В каждой телеконференции проводится ряд дискуссий по конкретным темам.

4. **Чат** (chat - разговор) - это ресурс Интернет, на котором посредством программ и скриптов реализована возможность одновременного общения большого количества пользователей в реальном времени.

5. **ICQ** (от англ. I Seek You – я ищу тебя)- это программа, реализующая функции Интернет-пейджера. ICQ в русском наречии часто именуемая "Аська". ICQ – это очень быстрый и наиболее популярный способ on-line

### 6. Программа пересылки файлов Ftp

Перемещает копии файлов с одного узла Интернет на другой в соответствии с протоколом **FTP** (File Transfer Protocol — "протокол передачи файлов"). В Интернет имеется более 10 Терабайт бесплатных файлов и программ.

### 7. Программа удалённого доступа Telnet.

Позволяет входить в другую вычислительную систему, работающую в Интернет, с помощью протокола **TELNET**. Эта программа состоит из двух компонент: программы-клиента, которая выполняется на компьютере-клиенте, и программы-сервера, которая выполняется на компьютере-сервере.

Telnet — простое и поэтому универсальное средство связи в Интернет.

8. **Поисковая система** – это комплекс программ и мощных компьютеров, автоматически просматривающих ресурсы Интернет, которые они могут найти, и индексирующих их содержание. Поисковые системы могут отличаться по эффективности поиска, по языку поиска (русский, английский и др.) и по некоторым другим возможностям. Например, одни поисковые системы находят информацию только в виде Web-страниц, другие могут просматривать и группы новостей, и файловые серверы. Результатом поиска являются гиперссылки на документы, содержащие требуемую информацию.

Для поиска информации в российских информационных ресурсах:

**Яндекс** (<http://www.yandex.ru/>).

**Рамблер** (<http://www.rambler.ru/>);

**Апорт** (<http://www.aport.ru/>).

## 8.9. Вопросы для самопроверки по теме 8

*Задание № 1.* Расшифруйте аббревиатуры LAN, MAN, WAN.

*Задание № 2.* Как называется иерархическая система назначения уникальных текстовых имён каждому компьютеру, находящемуся в сети?

*Задание № 3.* Укажите, какая часть электронного адреса ресурса описывает путь к файлу, расположенному на сервере:  
<http://www.yo.google.com/inf02000/01-02/det123.html>

*Задание № 4.* Укажите тип связи, который на сегодня является наиболее защищённым от несанкционированного доступа: оптоволоконный кабель, телефонный кабель (витая пара), электромагнитные волны.

*Задание № 5.* Укажите, какие термины не относятся к базовым топологиям сетей: линейная (шинная), кольцевая, снежинка, полносвязанная (сетевая).

*Задание № 6.* Укажите среди приведенных терминов, те, которые обозначают беспроводную связь: Ethernet, Wi-Fi, IrDa, FDDI.

*Задание № 7.* Укажите, что в сетевой терминологии обозначают термины «сервер», «клиент».

*Задание № 8.* Укажите, что в сетевой терминологии обозначает термин «протокол».

*Задание № 9.* На сколько уровней разбиты протоколы сетевой модели OSI?

## **9. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ**

### **9.1. Этапы решения задач на компьютере**

1. *Постановка задачи:* – сбор информации о задаче; – описание исходных данных и конечных целей; – определение формы выдачи результатов.

2. *Анализ и исследование модели задачи:* – анализ существующих аналогов; – анализ технических и программных средств; – разработка математической модели; – разработка структур данных.

3. *Разработка алгоритма:* – выбор метода проектирования алгоритма; – выбор формы записи алгоритма (блок-схема, псевдокод и т. п.

4. *Программирование:* – выбор языка программирования;

5. *Отладка и тестирование:*  
– синтаксическая отладка: исправление ошибок в форме записи конструкций;

– отладка семантики и логической структуры: семантика

– система правил истолкования отдельных конструкций языка, например проверка правильности организации циклов, ветвлений и т. п., соответствия типов переменных в выражениях, логическая структура

– правильная последовательность обработки данных;

– тестовые расчёты и анализ результатов тестирования;  
– совершенствование программы. Деятельность, направленная на исправление ошибок в программной системе, называется *отладкой*. **Тестирование** – прогон отлаженной программы на эталонных вариантах исходных данных, для которых заранее известны результаты (см. п. 5.3).

6. *Анализ результатов тестирования* и, если нужно, уточнение модели и повторение п.п. 2–5.

7. *Сопровождение программы*: составление документации по математической модели, алгоритму, программе, набору тестов, использованию готовой программы и т. п.

## **9.2. Понятие алгоритма и его свойства. Способы задания алгоритмов**

Наша учеба, работа, личные дела - это каждодневное, ежечасное решение различных задач. Каждая задача требует для своего решения выполнения определенных действий. Многократно решая задачи, можно заметить, что необходимые действия должны выполняться в строго определенном порядке. В таких случаях принято говорить об алгоритме решения задач. Понятие алгоритма считается одним из древнейших. Оно возникло задолго до появления ЭВМ, но с развитием вычислительной техники его роль значительно возросла.

Происхождение понятия алгоритма связано с именем великого среднеазиатского ученого Аль Хорезми, жившего в 9 веке н.э. Им были сформулированы впервые правила выполнения четырех арифметических действий.

Алгоритм - это точная инструкция, а инструкции встречаются во всех областях человеческой деятельности. Однако не всякую инструкцию можно назвать алгоритмом. Решая задачу, человек часто не задумывается над тем, как он это делает, и порой, затрудняется записать последовательность выполняемых действий. Но для того, чтобы поручить решение задачи автоматическому устройству необходимо составить алгоритм с четким указанием последовательности действий. Чтобы автоматическое устройство могло решить задачу в соответствии с алгоритмом, оно должно понимать каждое указание алгоритма. Алгоритм применяется к искомому набору исходных величин, называемых аргументами. Цель исполнения алгоритма получение определенного результата, если в результате исполнения алгоритма не достигнута определенная цель, значит алгоритм либо неверен, либо не завершен.

*Алгоритм – это чётко определённая последовательность действий, описывающих процесс преобразования объекта из начального состояния в конечное с помощью понятной исполнителю последовательности команд.*

Из этого определения следует, что правильно составленный алгоритм характеризуется следующими свойствами:

1. **Универсальность (массовость)** - применимость алгоритма к различным наборам исходных данных.

2. **Дискретность** - процесс решения задачи по алгоритму разбит на отдельные действия.

3. **Однозначность** - правила и порядок выполнения действий алгоритма имеют единственное толкование.

4. **Конечность** - каждое из действий и весь алгоритм в целом обязательно завершаются.

5. **Результативность** - по завершении выполнения алгоритма обязательно получается конечный результат.

6. **Выполнимость** - результата алгоритма достигается за конечное число шагов.

Алгоритм считается правильным, если его выполнение дает правильный результат. Соответственно алгоритм содержит ошибки, если можно указать такие допустимые исходные данные или условия, при которых выполнение алгоритма либо не завершится вообще, либо не будет получено никаких результатов, либо полученные результаты окажутся неправильными.

На практике наиболее распространены следующие формы представления алгоритмов:

- **словесная** (запись на естественном языке);
- **графическая** (изображения из графических символов);
- **псевдокоды** (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- **программная** (тексты на языках программирования).

Рассмотрим графический способ представления алгоритмов

Графический способ представления алгоритмов является более компактным и наглядным по сравнению с другими.

**При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий.**

Такое графическое представление называется схемой алгоритма или **блок-схемой**. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т.п.) соответствует геометрическая фигура, представленная в виде **блочного символа**. Блочные символы соединяются **линиями переходов**, определяющими очередность выполнения действий.

Для графического представления алгоритмов компьютерных программ используют специальные обозначения. Основные из них:

Название символа	Обозначение и пример заполнения	Пояснение
Процесс		Вычислительное действие или последовательность действий
Решение		Проверка условий
Модификация		Начало цикла
Предопределенный процесс		Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод		Ввод-вывод в общем виде
Пуск-останов		Начало, конец алгоритма, вход и выход в подпрограмму
Документ		Вывод результатов на печать

Блок **"процесс"** применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок **"решение"** используется для обозначения переходов управления по условию. В каждом блоке "решение" должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок **"модификация"** используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.

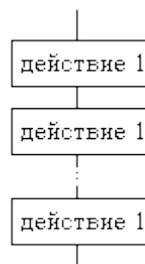
Блок **"предопределенный процесс"** используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам

В зависимости от последовательности выполнения действий в алгоритме выделяют алгоритмы линейной, разветвленной и циклической структуры.

**Линейный вычислительный процесс.** Образуется последовательностью действий, следующих одно за другим:

алгоритмический язык      Язык блок-схем

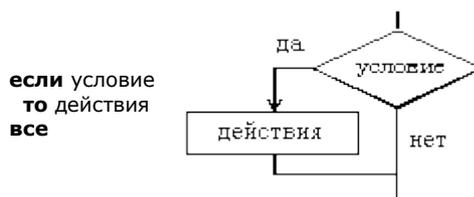
действие 1  
действие 2  
.....  
действие n



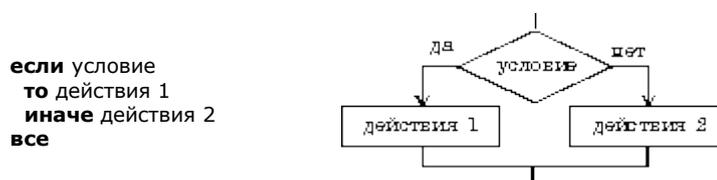
**Разветвляющийся вычислительный процесс.** Обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран. Структура **ветвление** существует в четырех основных вариантах:

- если—то;
- если—то—иначе;
- выбор;
- выбор—иначе.

### 1. если—то

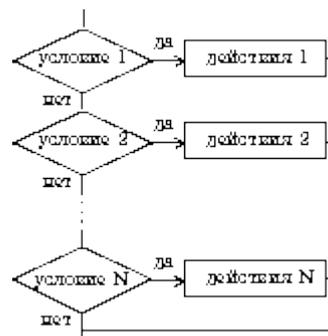


### 2. если—то—иначе



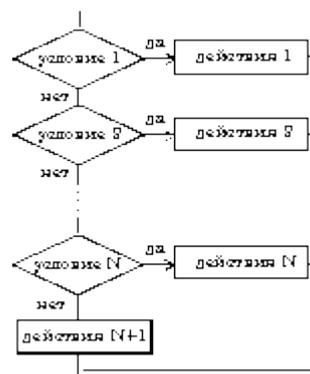
### 3. выбор

**выбор**  
**при** условие 1: действия 1  
**при** условие 2: действия 2  
 .....  
**при** условие N: действия N  
**все**

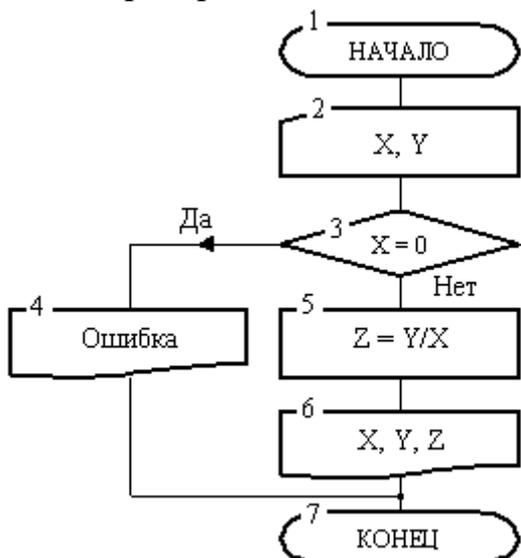


### 4. выбор—иначе

**выбор**  
**при** условие 1: действия 1  
**при** условие 2: действия 2  
 .....  
**при** условие N: действия N  
**иначе** действия N+1  
**все**



Рассмотрим решение задачи нахождения значения функции  $z = y/x$ .



**Циклический вычислительный процесс.** Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла. Основные разновидности циклов представлены в таблице:

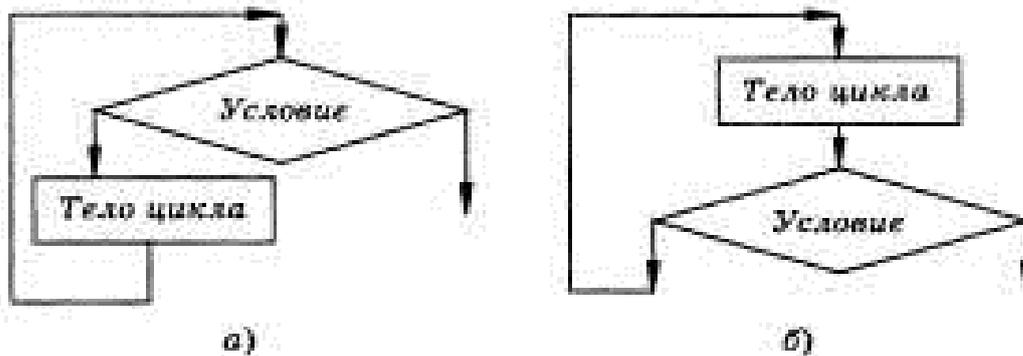


Рис. 6. Цикл с условием

а) **Цикл с предусловием** – проверяется условие и, если оно выполняется, то тело цикла (тц) повторяется, если нет – происходит переход к действию, следующему за телом цикла. Если условие не выполняется при первой проверке, то тело цикла не выполняется ни одного раза.

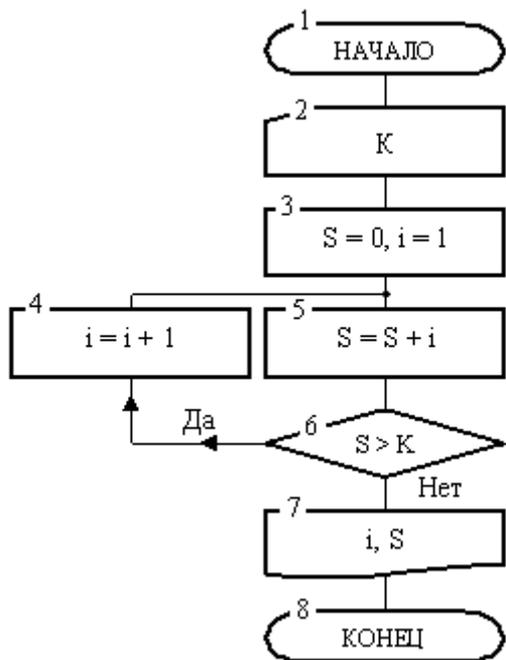
б) **Цикл с постусловием** – тело цикла (тц) выполняется, затем проверяется условие и, если оно не выполняется, то тело цикла повторяется, если выполняется, то происходит переход к действию, следующему за телом цикла. В этом варианте тело цикла выполняется хотя бы один раз всегда.

**Цикл со счётчиком** – выполняется заранее определённое количество раз:



Структура цикла

Рассмотрим пример алгоритма с циклом, имеющим наперед неизвестное количество проходов. Для этого решим следующую задачу. Указать наименьшее количество членов ряда натуральных чисел 1, 2, 3, ..., сумма которых больше числа К



### 9.3. Классификация языков программирования

На заре компьютерной эры машинный код был единственным средством общения человека с компьютером. Огромным достижением создателей языков программирования было то, что они сумели заставить сам компьютер работать переводчиком с этих языков на машинный код.

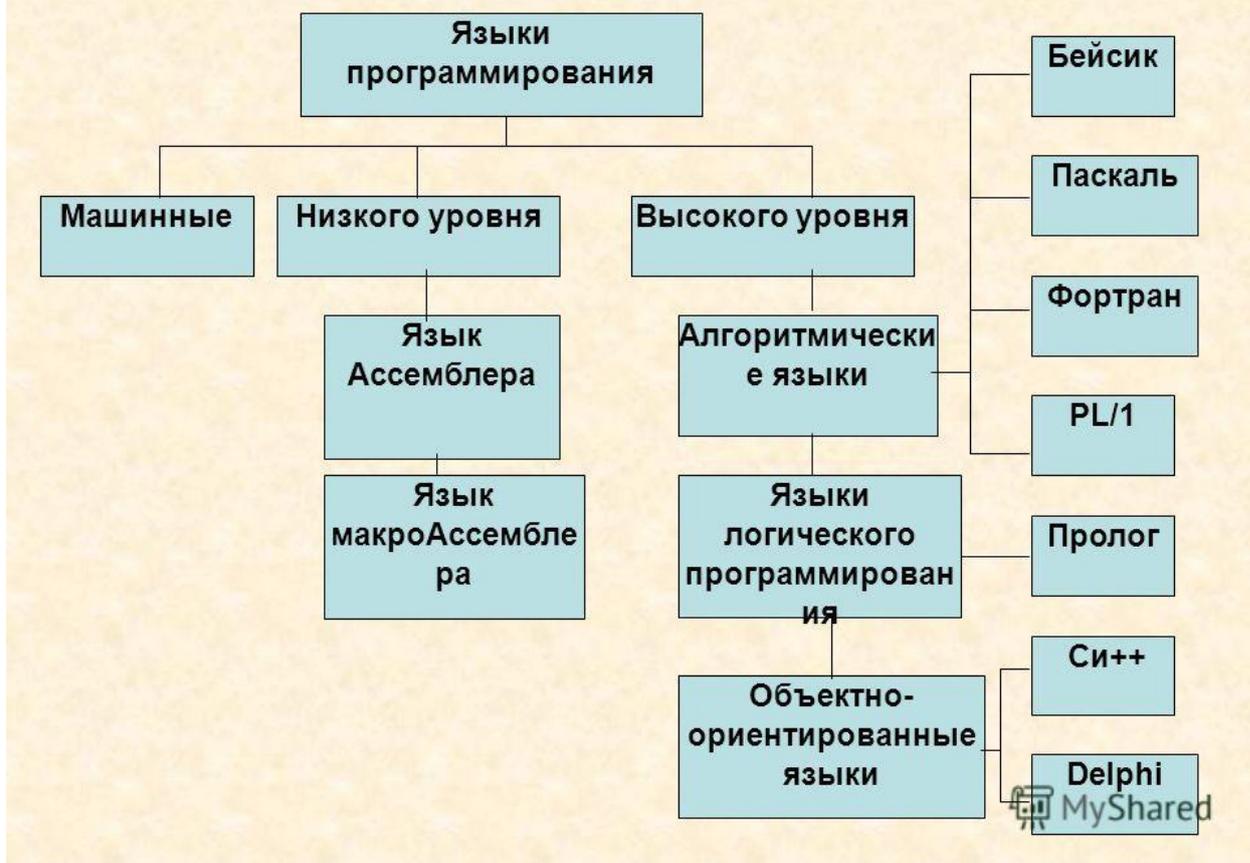
Существующие языки программирования можно разделить на две группы: процедурные и не процедурные

Процедурные (или алгоритмические) программы представляют из себя систему предписаний для решения конкретной задачи. Роль компьютера сводится к механическому выполнению этих предписаний.

Процедурные языки разделяют на языки низкого и высокого уровня.

Разные типы процессоров имеют разные наборы команд. Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется *языком программирования низкого уровня*. Имеется в виду, что операторы языка близки к машинному коду и ориентированы на конкретные команды процессора.

## Классификация и примеры языков программирования:



Языки низкого уровня (машинно-ориентированные) позволяют создавать программы из машинных кодов, обычно в шестнадцатеричной форме. С ними трудно работать, но созданные с их помощью высококвалифицированным программистом программы занимают меньше места в памяти и работают быстрее. С помощью этих языков удобнее разрабатывать системные программы, драйверы (программы для управления устройствами компьютера), некоторые другие виды программ.

Языком низкого уровня (машинно-ориентированным) является *Ассемблер*, который просто представляет каждую команду машинного кода, но не в виде чисел, а с помощью условных символьных обозначений, называемых *мнемониками*.

С помощью языков низкого уровня создаются очень эффективные и компактные программы, так как разработчик получает доступ ко всем возможностям процессора.

*Языки программирования высокого уровня* значительно ближе и понятнее человеку, нежели компьютеру. Особенности конкретных компьютерных архитектур в них не учитываются, поэтому создаваемые программы на уровне исходных текстов легко переносимы на другие платформы, для которых создан транслятор этого языка. Разрабатывать программы на языках высокого уровня с помощью понятных и мощных команд значительно проще, а ошибок при создании программ допускается гораздо меньше.

Основное достоинство алгоритмических языков высокого уровня - возможность описания программ решения задач в форме, максимально удобной для восприятия человеком. Но так как каждое семейство ЭВМ имеет свой собственный, специфический внутренний (машинный) язык и может выполнять лишь те команды, которые записаны на этом языке, то для перевода исходных программ на машинный язык используются специальные программы-трансляторы.

Работа всех трансляторов строится по одному из двух принципов: интерпретация или компиляция.

Классическое процедурное программирование требует от программиста детального описания того, как решать задачу, т. е. формулировки алгоритма и его специальной записи. При этом ожидаемые свойства результата обычно не указываются. Основные понятия языков этих групп – оператор и данные. При процедурном подходе операторы объединяются в группы – процедуры. Структурное программирование в целом не выходит за рамки этого направления, оно лишь дополнительно фиксирует некоторые полезные приемы технологии программирования.

Принципиально иное направление в программировании связано с методологиями (иногда говорят «парадигмами») не процедурного программирования. К ним можно отнести объектно-ориентированное и декларативное программирование. Объектно-ориентированный язык создает окружение в виде множества независимых объектов. Каждый объект ведет себя подобно отдельному компьютеру, их можно использовать для решения задач как «черные ящики», не вникая во внутренние механизмы их функционирования. Из языков объектного программирования, популярных среди профессионалов, следует назвать прежде всего Си++, для более широкого круга программистов предпочтительны среды типа Delphi и VisualBasic.

При использовании декларативного языка программист указывает исходные информационные структуры, взаимосвязи между ними и то, какими свойствами должен обладать результат. При этом процедуру его получения («алгоритм») программист не строит (по крайней мере, в идеале). В этих языках отсутствует понятие «оператор» («команда»). Декларативные языки можно подразделить на два семейства – логические (типичный представитель – Пролог) и функциональные (Лисп).

Охарактеризуем наиболее известные языки программирования.

1. **Фортран** (FORmulaTRANslatingsystem – система трансляции формул); старейший и по сей день активно используемый в решении задач математической ориентации язык. Является классическим языком для программирования на ЭВМ математических и инженерных задач

2. **Бейсик** (Beginner'sAll-purposeSymbolicInstructionCode – универсальный символический код инструкций для начинающих); несмотря на многие недостатки и изобилие плохо совместимых версий – самый

популярный по числу пользователей. Широко употребляется при написании простых программ.

3. **Алгол** (ALGOrithmicLanguage – алгоритмический язык); сыграл большую роль в теории, но для практического программирования сейчас почти не используется.

4. **ПЛ/1** (PL/1 ProgrammingLanguage – язык программирования первый); многоцелевой язык; сейчас почти не используется.

5. **Паскаль** (Pascal – назван в честь ученого Блеза Паскаля); чрезвычайно популярен как при изучении программирования, так и среди профессионалов. Создан в начале 70-х годов швейцарским ученым Никлаусом Виртом. Язык Паскаль первоначально разрабатывался как учебный, и, действительно, сейчас он является одним из основных языков обучения программированию в школах и вузах. Однако качества его в совокупности оказались столь высоки, что им охотно пользуются и профессиональные программисты. Не менее впечатляющей, в том числе и финансовой, удачи добился Филип Кан, француз, разработавший систему Турбо-Паскаль. Суть его идеи состояла в объединении последовательных этапов обработки программы – компиляции, редактирования связей, отладки и диагностики ошибок – в едином интерфейсе. Версии Турбо-Паскаля заполнили практически все образовательные учреждения, программистские центры и частные фирмы. На базе языка Паскаль созданы несколько более мощных языков (Модула, Ада, Дельфи).

6. **Кобол** (COmmonBusinessOrientedLanguage – язык, ориентированный на общий бизнес); в значительной мере вышел из употребления. Был задуман как основной язык для массовой обработки данных в сферах управления и бизнеса.

7. **АДА**; является языком, победившим (май 1979 г.) в конкурсе по разработке универсального языка, проводимым Пентагоном с 1975 году. Разработчики – группа ученых во главе с Жаном Ихбиа. Победивший язык окрестили АДА, в честь Огасты Ады Лавлейс. Язык АДА – прямой наследник языка Паскаль. Этот язык предназначен для создания и длительного (многолетнего) сопровождения больших программных систем, допускает возможность параллельной обработки, управления процессами в реальном времени и многое другое, чего трудно или невозможно достичь средствами более простых языков.

8. **Си** (C – «си»); широко используется при создании системного программного обеспечения. Наложил большой отпечаток на современное программирование (первая версия – 1972 г.), является очень популярным в среде разработчиков систем программного обеспечения (включая операционные системы). Си сочетает в себе черты как языка высокого уровня, так и машинно-ориентированного языка, допуская программиста ко всем машинным ресурсам, чего не обеспечивают такие языки, как Бейсик и Паскаль.

9. **Си++** (C++); объектно-ориентированное расширение языка Си, созданное Бьярном Страуструпом в 1980 году. Множество новых мощных возможностей, позволивших резко повысить производительность программистов, наложилось на унаследованную от языка Си определенную низкоуровневость.

10. **Дельфи** (Delphi); язык объектно-ориентированного «визуального» программирования; в данный момент чрезвычайно популярен. Созданный на базе языка Паскаль специалистами фирмы Borland язык Delphi, обладая мощностью и гибкостью языков Си и Си++, превосходит их по удобству и простоте интерфейса при разработке приложений, обеспечивающих взаимодействие с базами данных и поддержку различного рода работ в рамках корпоративных сетей и сети Интернет.

11. **Ява** (Java); платформенно-независимый язык объектно-ориентированного программирования, чрезвычайно эффективен для создания интерактивных веб-страниц. Этот язык был создан компанией Sun в начале 90-х годов на основе СИ++. Он призван упростить разработку приложений на основе Си++ путем исключения из него всех низкоуровневых возможностей.

12. **Лисп** (Lisp) – функциональный язык программирования. Ориентирован на структуру данных в форме списка и позволяет организовать эффективную обработку больших объемов текстовой информации.

13. **Пролог** (PROgramminginLOGic – логическое программирование). Главное назначение языка – разработка интеллектуальных программ и систем. Пролог – это язык программирования, созданный специально для работы с базами знаний, основанными на фактах и правилах (одного из элементов систем искусственного интеллекта). В языке реализован механизм возврата для выполнения обратной цепочки рассуждений, при котором предполагается, что некоторые выводы или заключения истинны, а затем эти предположения проверяются в базе знаний, содержащей факты и правила логического вывода. Если предположение не подтверждается, выполняется возврат и выдвигается новое предположение. В основу языка положена математическая модель теории исчисления предикатов.

Языки программирования для Интернета:

1. **HTML**. Общеизвестный язык для оформления документов. Он очень прост и содержит элементарные команды форматирования текста, добавления рисунков, задания шрифтов и цветов, организации ссылок и таблиц.

2. **PERL**. Он задумывался как средство эффективной обработки больших текстовых файлов, генерации текстовых отчетов и управления задачами. По мощности Perl значительно превосходит языки типа Си. В него введено много часто используемых функций работы со строками, массивами, управление процессорами, работа с системной информацией.

3. **Tcl/Tk**. Этот язык ориентирован на автоматизацию рутинных процессов и состоит из мощных команд. Он независим от системы и при этом позволяет создавать программы с графическим интерфейсом.

4. **VRML.** Создан для организации виртуальных трехмерных интерфейсов в Интернете. Он позволяет описывать в текстовом виде различные трехмерные сцены, освещение и тени, текстуры.

Выбор языка программирования зависит от многих факторов: назначения, удобства написания исходных программ, эффективности получаемых объектных программ и т. п. Разнотипность решаемых компьютером задач и определяет многообразие языков программирования.

**Заключение.** Изучение вопросов эволюции языков программирования призвано облегчить программисту выбор языка для решения определенных задач. Однако следует осознавать, что не все они полиглоты и не нужно изучать все существующие языки программирования – достаточно изучать по одному языку каждого класса по мере необходимости, так как в процессе эволюции все языки одного класса сближаются. И помните главное: лучший язык тот, который знаешь в совершенстве.

#### **9.4. Основные понятия алгоритмических языков. Алфавит. Синтаксис. Семантика**

Обычный разговорный язык состоит из четырех основных элементов: *символов, слов, словосочетаний и предложений.* Алгоритмический язык содержит подобные элементы, только слова называют *элементарными конструкциями,* словосочетания – *выражениями,* предложения – *операторами.* Алгоритмический язык (как и любой другой язык), образуют три его составляющие: алфавит, синтаксис и семантика.

**Алфавит** – фиксированный для данного языка набор символов (букв, цифр, специальных знаков и т.д.), которые могут быть использованы при написании программы.

**Синтаксис** – правила использования символов алфавита в специальных конструкциях, с помощью которых составляется алгоритм.

**Семантика** – система правил толкования конструкций языка. Таким образом, программа составляется с помощью соединения символов алфавита в соответствии с синтаксическими правилами и с учетом правил семантики. Программа, написанная на алгоритмическом языке, называется исходным модулем. Для ее трансляции можно использовать два типа технологий: интерпретацию или компиляцию. Трансляторы, реализующие эти технологии, называются соответственно интерпретаторами и компиляторами. При интерпретации транслятор переводит в машинные коды очередную строку исходного модуля, сразу же выполняет её и переходит к обработке следующей строки, не создавая законченного исполняемого модуля всей программы в целом. Такой тип трансляции удобен на этапе отладки или тестирования программы. При компиляции трансляция разбивается на два этапа. На первом, компилятор создает объектный модуль – промежуточную программу, в которой конструкции исходного модуля переведены в машинные коды, но вместо реальных адресов данных и работающих с ней подпрограмм, используются относительные адреса и не добавлены

служебные подпрограммы, необходимые для её работы. На втором этапе программа-сборщик заменяет в объектном модуле относительные адреса на абсолютные и подключает к нему необходимые для работы стандартные программы. В результате получается исполняемый модуль – программа, написанная в машинных кодах и полностью готовая к работе. Этот режим трансляции удобен для расчётов по уже отлаженной и протестированной программе, так как исполняемый модуль можно сохранить в памяти и при необходимости провести расчёты с другим набором исходных данных, не повторяя трансляцию.

В настоящее время для большинства алгоритмических языков разработаны интегрированные системы программирования (Бейсик, Паскаль, Си и т. п.). Они включают в себя следующие компоненты:

- текстовый редактор для написания программы;
- библиотеки стандартных программ;
- средства отладки;
- справочную службу;
- компилятор и/или интерпретатор.

## 9.5. Язык программирования PascalABC

**PascalABC.NET** — это язык программирования нового поколения, включающий классический Паскаль, большинство возможностей языка Delphi, а также ряд собственных расширений. Он реализован на платформе Microsoft.NET и содержит все современные языковые средства: классы, перегрузку операций, интерфейсы, обработку исключений, обобщенные классы и подпрограммы, сборку мусора, лямбда-выражения, средства параллельного программирования.

PascalABC.NET является мультипарадигменным языком: на нём можно программировать в структурном, объектно-ориентированном и функциональном стилях.

PascalABC.NET — это также простая и мощная интегрированная среда разработки, поддерживающая технологию IntelliSense, содержащая средства автоформатирования, встроенный отладчик и встроенный дизайнер форм. Кроме того, консольный компилятор PascalABC.NET функционирует на Linux и MacOS под Mono.

## 9.6. Структура программы языка Pascal

Программа в Pascal состоит из двух разделов

**1. Раздел описаний** – здесь указываются все необходимые компоненты для вашей программы.

Program	Название программы
Uses	Раздел подключения библиотек
Const	Раздел описания постоянных величин
Var	Раздел описания переменных
Function procedure	Раздел описания функций и процедур.

## 2. Раздел операторов.

В этом разделе производятся все вычисления и выполняются все операции.

**Begin** - начало программы

Набор операторов

**End.** - конец программы

Хочется обратить внимания на то, что в конце каждой строки ставиться

**Точка с запятой.**

Для написания любой программы нам понадобятся переменные величины – ячейки памяти компьютера, в которых будут храниться какие-то значение.

**Тип** определяет множество значений, которые могут принимать объекты программы (константы и переменные), а также совокупность операций, допустимых над этими значениями.

**Целые и вещественные** типы данных предназначены для представления числовых данных. **Целый** тип в языке TurboPascal – это интервал целых чисел (таблица 1). Операции над этими числами определены лишь тогда, когда исходные данные (операнды) и результат лежат в этом интервале.

Таблица 1 – Целочисленные типы данных

<i>Название целого типа</i>	<i>Диапазон возможных значений</i>	<i>Память, байт</i>
<b>byte</b> (байтовый)	0 – 255	1
<b>shortint</b> (короткий целый)	-128 – 127	1
<b>integer</b> (целый)	-32 768 – 32 767	2
<b>word</b> (слово)	0 – 65 535м	2
<b>longint</b> (длинный целый)	-2 147 483 648 – 2 147 483 687	4

**Вещественные** числа представляются конечным множеством значений (таблица 2).

Таблица 2 – Вещественные типы данных

Название вещественного типа	Диапазон возможных значений (плюс-минус)	Количество значащих чисел	Память, байт
<b>single</b> (с одинарной точностью)	1,5e-45 – 3,4e38	7 – 8	4
<b>real</b> (вещественный)	2,9e-39 – 1,7e38	11 – 12	6
<b>double</b> (с двойной точностью)	5,0e-324 – 1,7e308	15 – 16	8
<b>extended</b> (с повышенной точностью)	3,4e-4932 – 1,1e4932	19 – 20	10
<b>comp</b> (сложный)	-2e63+1 – 2e63-1	19 – 20	8

**Логический (boolean)** тип данных имеет всего два значения: true (да – истина, 1) и false (нет, ложь, 0).

**Строковый тип данных (string)** – это тип данных, состоящий из последовательности символов. Каждый символ занимает 1 байт памяти. Количество символов в строке называется её длиной. Длина строки может находиться в диапазоне от 0 до 255. Строковые величины могут быть константами и переменными. Строковая константа есть последовательность символов, заключенная в апострофы.

Например, 'Язык программирования TurboPascal', '12345', '' – пустая строка.

**Символьный тип данных (char)** – это тип данных, состоящий из одного символа (знака, буквы, кода) в определённой кодировке.

Например: 'ж', 's', '№', '\*', '\_' – пробел.

### **Ввод и вывод данных**

**Ввод данных** – это передача исходных данных программы в оперативную память компьютера для последующей обработки программой.

**Read (x1, x2, ..., xN);**

**Readln (x1, x2, ..., xN);**

где x1, x2, ... – список ввода, содержащий имена переменных допустимых типов данных (integer, real, char, string). Эти значения вводятся с клавиатуры и отображаются на экране.

**Вывод данных** – это передача данных после обработки из оперативной памяти на внешнее устройство (экран, принтер, файл на диске).

**Write (y1, y2, ..., yN);**

**Writeln (y1, y2, ..., yN);**

где y1, y2, ... – список вывода, предназначенный для вывода констант различных типов, значений переменных и выражений.

Процедура вывода **Write** и **Writeln** отличаются тем, что после вывода последней переменной из списка курсор автоматически переходит в начало новой строки (таблица 3).

Таблица 3 – Форматы вывода данных

<i>Значение</i>	<i>Выражение</i>	<i>Результат</i>
3и 45	Write ( A, B )	345 (с позиции курсора)
3 и 45	Write ('A=' , A , '*B=' , B)	A=3*B=45
3и 45	Write ( A : 4, B : 4)	***3**45 (крайняя правая позиция)
3и 45	Write ('A=' , A:4, '*B=' , B:4)	A=***3*B=**45
3и 45	Write (A:4); Write (B:4);	***3**45
3и 45	Writeln (A:4); Writeln (B:4);	***3 **45
234,36	Write ( A )	2.343600000E+02 (с плавающей точкой)
234,36	Write ( A : 8 : 2 )	**234.36 (с фиксированной точкой)
234,36	Write ( A : 8 : 4 )	234.3600
234,36	Write ( A : 10 : 1 )	*****234.4

Примечание: \* означает пробел

**Общий вид записи:**

**Write(R:m:n);**

**R** – действительное число;

**m** – количество позиций, отводимых для числа;

**n** – количество позиций, отводимых для дробной части. Задание 1.

Запись математических выражений на алгоритмическом языке

$\frac{1,34 a^3 + b^3 }{bc} e^{a^2 - b^2}$	$1.34*abs(exp(3*ln(a))+exp(3*ln(b)))/(b*c)*exp(a*a - b*b)$
--	--

Таблица 4 – арифметические функции

<i>Математика</i>	<i>TurboPascal</i>	<i>Расшифровка</i>
$a^2$	<i>sqr(a)</i>	Квадрат
$\sqrt{a}$	<i>sqrt(a)</i>	Корень
$ a $	<i>abs(a)</i>	Модуль
$ln(a)$	<i>ln(a)</i>	Натуральный логарифм
$\pi$	<i>pi</i>	Число пи
$e^a$	<i>exp(a)</i>	Экспонента

<i>Математика</i>	<i>TurboPascal</i>	<i>Расшифровка</i>
$a^n$	$exp(n*ln(a))$	n-ая степень
$sin(a)$	$sin(a)$	Синус (в радианах)
$cos(a)$	$cos(a)$	Косинус (в радианах)
$tg(a)$	$sin(a) / cos(a)$	Тангенс (в радианах)
$ctg(a)$	$cos(a) / sin(a)$	Котангенс (в радианах)
$arctg(a)$	$arctan(x)$	Арктангенс (в радианах)
--	$a \text{ div } b$	Деление без остатка
--	$a \text{ mod } b$	Остаток от деления
--	<i>Random</i> (диапазон)	Псевдослучайное число [0,1]
--	<i>Random(x)</i>	Псевдослучайное число [0,x]

### Цикл с использованием операторов условного (IF) и безусловного переходов (GOTO)

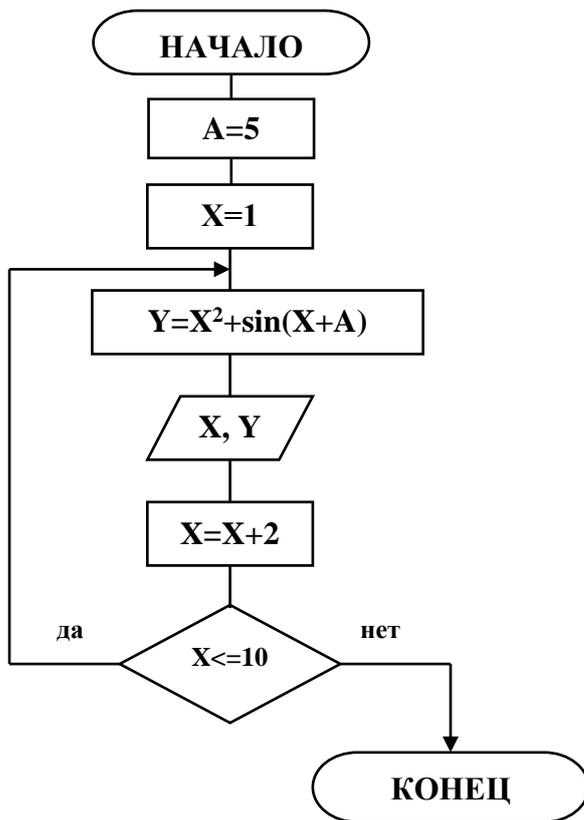
Если в программе после выполнения очередного оператора необходимо выполнить не следующий по порядку, а другой, помеченный для этого меткой, используется оператор безусловного перехода **GOTO** («переход на»). Переход возможен только в пределах блока.

**Форма записи оператора безусловного перехода:**

**GOTO** «Имя метки»;

«Имя метки» – может быть идентификатор или целые числа от **0** до **9999**. Объявление меток проводится перед объявлением констант после заголовка программы. Объявление начинается с ключевого слова **LABEL**, затем указывается имя метки.

**Пояснение:** 50 – имя метки.



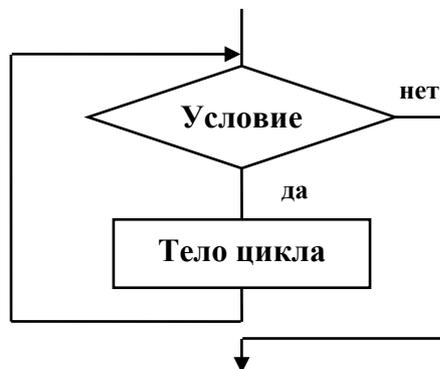
```

ProgramName1;
Label 50;      {раздел описания
меток}
Var
X, A: integer; Y: real;
Begin
A:=5;
X:=1;
50:  Y:=X*X+sin(X+A);
Writeln ('X=',X, '
Y=',Y:4:1);
X:=X+2;
if x<=10 then goto 50;
readln;
end.
  
```

## 2 способ: Организация цикла с предусловием (WHILE)

### *Общий вид алгоритма цикла с предусловием*

Предписывает выполнять тело цикла до тех пор, пока выполняется условие, записанное после слова **While**.



*Общий вид записи:*

**While** «Условие»**do**

«Тело цикла»;

«Условие» – выражение логического типа;

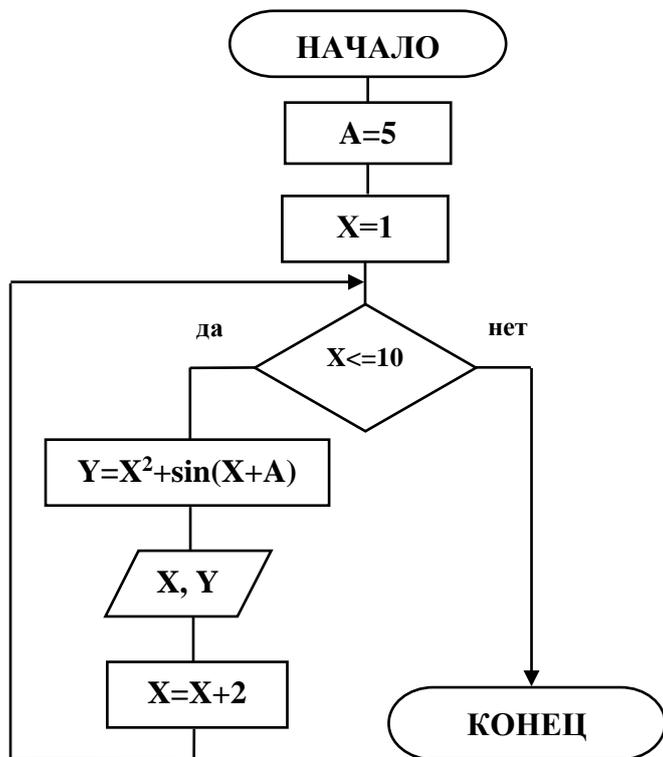
«Тело цикла» – простой или составной оператор.

В том случае, если в теле цикла имеется более одного оператора, в качестве оператора может выступать составной оператор с операторными скобками **begin...end**.

Перед каждым выполнением тела цикла вычисляется значение выражения условия, если результат истина, то тело цикла выполняется. Если

результат ложь, то происходит выход из цикла и переход к первому оператору после **While** оператора.

**While, do** – зарезервированные слова («Пока [выполняется условие]», «делать»).



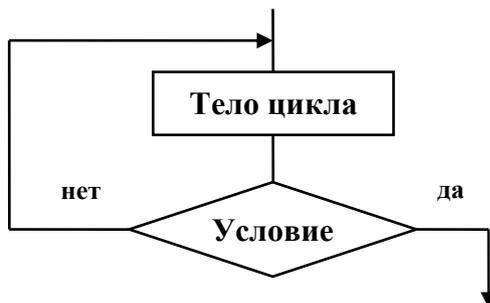
```

Program Name2;
Var
X,A: integer; Y: real;
Begin
A:=5;
X:=1;
While X<=10 do begin
  Y:=X*X+sin(X+A);
  Writeln ('X=',X,',
  Y=',Y:4:1);
  X:=X+2;
end;
readln;
end.
  
```

**3 способ: Организация цикла с постусловием (REPEAT...UNTIL)**

*Общий вид алгоритма цикла с постусловием*

Выполняется до тех пор, пока логическое выражение ложно. Условие нужно задать так, чтобы выйти из цикла.



*Общий вид записи:*

**Repeat**

«Тело цикла»;

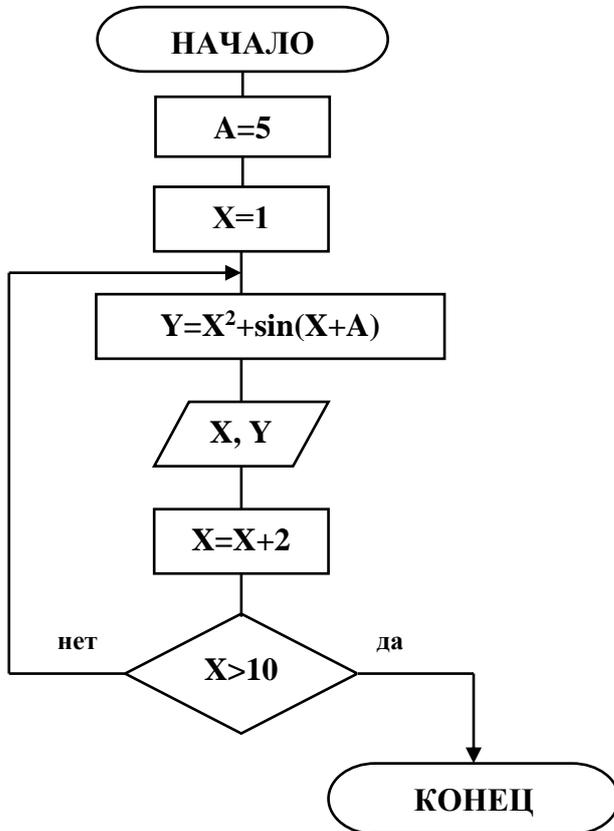
**until** «Условие окончания цикла»;

«Тело цикла» – произвольная последовательность операторов;

«Условие» – выражение логического типа.

Процесс завершается тогда, когда после очередного выполнения заданной последовательности операторов тела цикла условие примет (впервые) значение истина. Проверка условия производится после выполнения тела цикла.

**Repeat...until** – зарезервированные слова («Повторять», «пока не [выполнится условие]»).



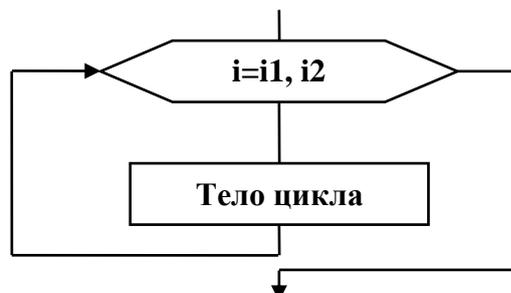
```

Program Name3;
Var
X, A: integer; Y: real;
Begin
A:=5;
X:=1;
  repeat
    Y:=X*X+sin(X+A);
    Writeln ('X=',X,', Y=',Y:4:1);
    X:=X+2;
  until X>10;
readln;
end.
  
```

#### 4 способ: Организация цикла с параметром(FOR)

##### *Общий вид алгоритма цикла с параметром*

Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне между начальным и конечным значениями. Используется в том случае, если параметр цикла имеет **целочисленный** тип и значение параметра увеличивается (уменьшается) на **1**.



- i** – параметр цикла
- i1** – начальное значение,
- i2** – конечное значение

**Общий вид записи:**

**for** «Параметр цикла»:=«Начальное значение» **to** (downto) «Конечное значение»  
**do**

«Тело цикла»;

В качестве параметра цикла можно использовать имя простой переменной целого типа, в качестве начального, конечного значений можно использовать константы, переменные, а также арифметические выражения только целого или перечислимого типа.

**for, to, downto, do** – зарезервированные слова («для», «до», «уменьшить до», «выполнить»).

**Порядок выполнения действий:**

До разработки алгоритма решения задачи необходимо вычислить количество повторений цикла **K**.

$$K = \frac{x_{\text{кон}} - x_{\text{нач}}}{\Delta x} + 1; K - \text{количество повторений цикла (целое число)}$$

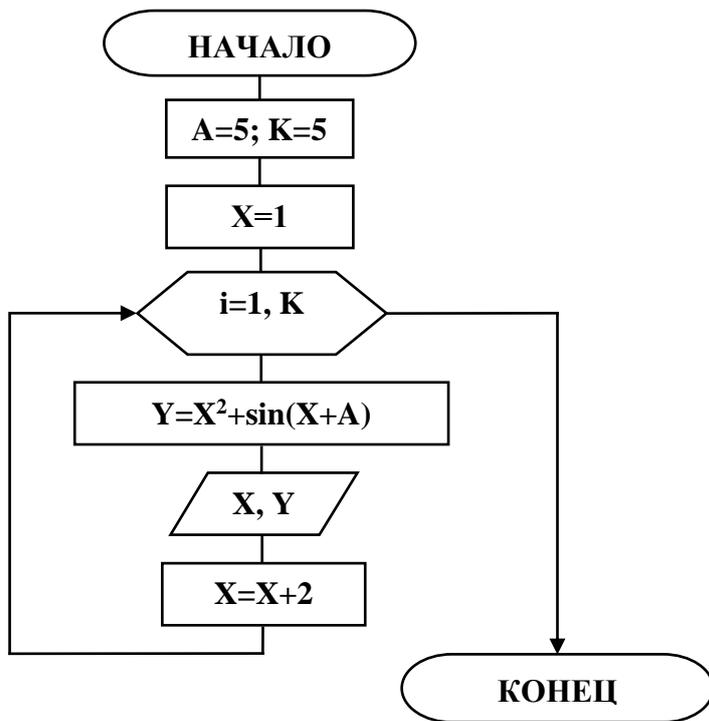
$$x_{\text{нач}} = 1; x_{\text{кон}} = 10; \Delta x = 2$$

$$K = \frac{10 - 1}{2} + 1 = 4,5 + 1 = 5,5$$

$K = 5$  – т.к. если  $K = 6$ , то  $x = 12$  и не принадлежит диапазону  $[1; 10]$

**K** – в данной программе будет означать конечное значение цикла.

- 1) Присвоение значения переменной **A**.
- 2) Присвоение переменной **K** значения, полученного до разработки алгоритма решения задачи.
- 3) Присвоение значения **1** переменной **X** (начальное значение).
- 4) Организация цикла с параметром по переменной **i** (целое число).
- 5) Вычисление значения функции **Y**.
- 6) Вывод значения функции **Y**.
- 7) Добавление шага к переменной **X**.



```

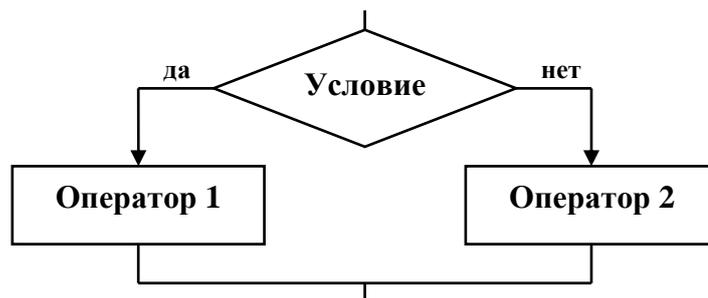
Program Name4;
Var
X, A, K i: integer; Y: real;
Begin
A:=5; K:=5;
X:=1;
for i:=1 to K do begin
  Y:=X*X+sin(X+A);
  Writeln ('X=',X,
';Y=',Y:4:1);
  X:=X+2;
end;
readln;
end.
  
```

#### Задание 5. Разветвление в цикле

Разветвление в цикле означает, что тело цикла содержит разветвляющийся процесс.

**Разветвляющийся процесс** – это процесс, в котором вычисления могут выполняться разными путями в зависимости от выполнения или невыполнения некоторых условий. Обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

#### *Общий вид алгоритма разветвляющегося процесса*



#### *Общий вид записи с условным оператором:*

**If** «Условие»**then** «Оператор\_1»**else** «Оператор\_2»;

«Условие» – логическое выражение.

**if, then, else** – зарезервированные слова («если», «то», «иначе»).

*Условие задачи:*

Найдите значения функции  $F = B \cdot D + \ln D$ ,

если

$$D = \begin{cases} \frac{A+X}{X}, & \text{при } X < 2,3 \\ \frac{A+B}{X+1}, & \text{при } 2,3 \leq X < 5 \\ A \cdot X^2 + B \cdot X, & \text{при } X \geq 5 \end{cases}$$

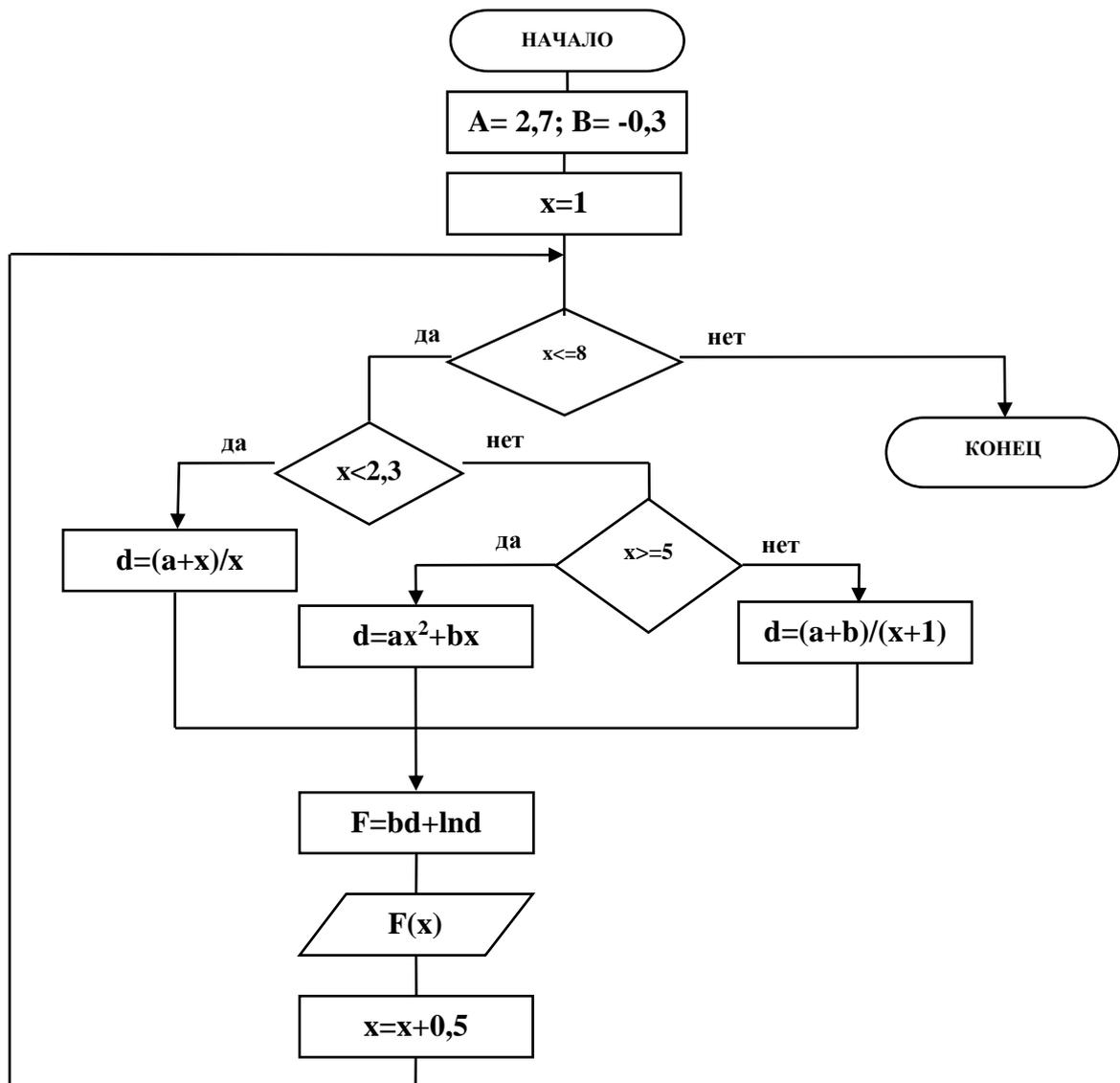
$$A = 2,7; B = -0,3$$

$$\text{где } X \in [1;8], \Delta X = 0,5$$

**Порядок выполнения действий:**

1. Присвоение значений переменным **A** и **B**.
2. Организация цикла по переменной **X**.
3. Вычисление функции **D** в зависимости от условия.
4. Вычисление значений функции **F**.
5. Вывод значения функции **F**.

1 способ: Цикл с предусловием (WHILE)



Program funk\_F1;

Var

a, b, d, x, F: real;

begin

a:= 2.7; b:= -0.3;

x:=1;

While x<=8 do begin

if x<2.3 then d:=(a+x)/x else

if x>=5 then d:=a\*x\*x+b\*x else

d:=(a+b)/(x+1);

F:=b\*d+ln(d);

writeln('F(',x:2:1, ')= ',F:6:2);

x:=x+0.5;

end;

readln;

end.

## Задание 6. Обработка одномерных массивов

**Одномерный массив** – это фиксированное количество элементов одного и того же типа, объединенных одним именем, где каждый элемент имеет свой номер.

Например, **A(8)** – одномерный массив, где **A** – имя массива, **8** – количество элементов.

**Общий вид записи:**

**A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub>....., A<sub>n</sub>** – это одномерный массив,  
где **A<sub>i</sub>** – элементы массива, **i=1, 2, 3.....n** – индекс представляет собой переменную целого типа. Запись элемента массива на языке Паскаль **A[i]**.

**Общий вид записи описания одномерного массива:**

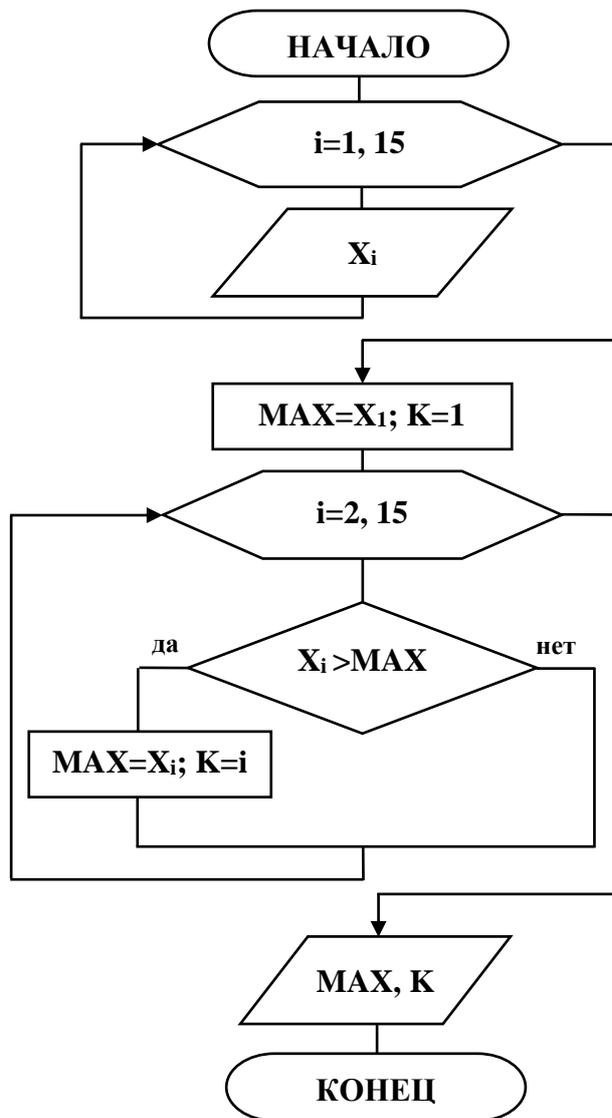
**Var** «Имя массива»: **Array[n1..n2] Of** «Тип элементов»;

**n1** – номер первого элемента;

**n2** – номер последнего элемента.

**Array,of** – зарезервированные слова («массив», «из»).

**Задача 1.** В массиве **X(15)** вычислите максимальный элемент массива и его порядковый номер.



### Обозначения:

**MAX** – максимальный элемент массива;  
**K** – порядковый номер максимального элемента массива.

ProgramMassiv1;

Var

{Описание используемых переменных}

K, i, MAX: integer;

{Описание массива X}

X: array [1..15] of integer;

begin

{Организация цикла для ввода элементов массива с клавиатуры}

for i:=1 to 15 do

readln (X[i]);

{Присвоение начальных значений максимальному элементу и его порядковому номеру}

MAX:=X[1]; K:=1;

{Организация цикла для поиска максимального элемента массива и его порядкового номера}

for i:=2 to 15 do

if X[i]>MAX then begin MAX:=X[i];

K:=i; end;

{Вывод результатов вычисления на экран}

writeln ('MAX=', MAX:2, ';K=', K);

readln; end.

## 9.7. Вопросы для самопроверки по теме 9

**Задание №1.** Укажите вариант описания, соответствующий циклу с постусловием:

1. пока условие истинно, выполнять оператор;
2. выполнять оператор, пока условие ложно;
3. выполнять оператор заданное число раз;
4. если условие истинно, выполнять оператор, иначе – остановиться.

**Задание №2.** Определите, сколько раз тело цикла выполняется в представленном фрагменте:

V := 10; D := 40

Начало цикла: пока D >= V

D := D – V

Конец цикла.

*Задание №3.* Определите вид вывода результатов в предложенном фрагменте программы:

X:=5

Z:=7

ВЫВОД ("X=",X," X=",Z,Z+X)

1. X=5 X=712
2. "X=",5," X=",7,12
3. X=5 X=7 12
4. "X=",X," X=",Z,Z+X

*Задание № 4.* Ошибка в форме записи программы – это \_\_\_\_\_ ошибка.

*Задание № 5.* К языкам программирования низкого уровня относят:

1. Basic;
2. Ассемблер;
3. Pascal;
4. Lisp.

*Задание № 6.* Алгоритмом является следующий документ:

1. правила техники безопасности;
2. инструкция по приготовлению питательной среды для микроорганизмов;
3. расписание занятий;
4. схема устройства принтера

*Задание № 7.* Алгоритмическая конструкция, в которой в зависимости от условий может выполняться либо одно действие, либо другое называется:

1. линейной;
2. ветвящейся;
3. циклической;
4. рекурсивной.

*Задание № 8.* Алгоритм должен обязательно соответствовать требованию:

1. открытость;
2. простота;
3. объективность;
4. массовость.

*Задание №9.* Непосредственное покомандное выполнение исходной программы на языке высокого уровня происходит в процессе:

1. интерпретации;
2. идентификации;
3. транзакции;
4. компиляции.

## ОТВЕТЫ НА ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

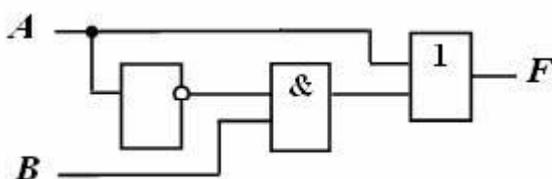
### 1. Основные понятия теории информации

№ вопроса	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Ответ	152 бит	28 байт	3	2	2	2	4	2	1	111111111	1Кб, 1Мб, 1Гб, 1Тб	120бит, 15байт	2	вербальная

### 2. Введение в алгебру логики

№ вопроса	1	2	3	4	5	7
Ответ	2	2	Простое ложное	3	2	3

Задание №6:



### 3. Технические средства реализации информационных процессов

№ вопроса	Ответ
1	1
2	1, 2, 3
3	4 поколение
4	Физический размер экрана и угол обзора
5	4
6	1
7	CD-ROM, CD-R, DVD-ROM, DVD-R, DVD-RAM
8	3
9	3
10	Процессор, ОЗУ, ПЗУ, Кэш-память, интерфейсные схемы шин, гнезда расширений (слотов), обязательные системные средства ввода/вывода.
11	По механизму выполнения печати
12	Устройство для соединения блоков компьютера с разными способами представления информации
13	1,4
14	Жесткий диск, гибкий диск, оптические диски, цифровые видеодиски, магнитооптические диски, стримеры

#### 4. Программные средства реализации информационных процессов

№ вопроса	1	2	3	4	5	6	7	8
Ответ	3	2,3,4,5	3	3	3	4	3	1

#### 5. Основные понятия информационного обеспечения

№ вопроса	1	2	3	4	5
Ответ	3	2	3	2	1

#### 6. Основы защиты информации

№ вопроса	1	2	3	4	5	6
Ответ	2,3,4,5,6	2	2	1,2,3,4	См. пункт 7.3	См. пункт 7.4

#### 7. Информационно-коммуникационные технологии

№ вопроса	Ответ
1	Local Areal Net, Metropolitan Areal Net, World Areal Net
2	Доменная система имён
3	inf02000/01-02
4	Оптоволоконный кабель
5	Снежинка
6	Wi-Fi, IrDa
7	Сервер – это компьютер, на котором хранится информация, необходимая разным пользователям или же выполняющий работы по поддержанию связи между компьютерами Рабочая станция (клиент) – это внутренний компьютер ЛВС
8	Сетевые протоколы – это наборы правил для обмена информацией в сети и разработки сетевого оборудования: HTTP, SMTP, FTP, DNS, POP3
9	7 уровней

#### 8. Алгоритмизация и программирование

№ вопроса	1	2	3	4	5	6	7	8	9
Ответ	2	4 раза	1	Синтаксическая ошибка	2	1,2,3	2	4	1

## СПИСОК ЛИТЕРАТУРЫ

1. Курс лекций по информатике: Учебное пособие / Аронова Т.В., Конюхова Г.П., Яшкина Е.Н. - М.: РГУФКСМиТ, 2011. – 254 с.
2. «Информатика: Операционная система Microsoft Windows»: учебное пособие для студентов бакалавриата / Сост. И.Г. Абышева, Н.В. Горбушина, А.Г. Семенова. - Ижевск: РИО ФГБОУ ВО Ижевская ГСХА, 2017. - 56 с.
3. «Информатика: практикум для студентов бакалавриата и специалитета» / Сост. И.Г. Абышева, Н.В. Горбушина, А.Г. Семенова. - Ижевск: РИО ФГБОУ ВО Ижевская ГСХА, 2017. – 54 с.
4. «Информатика: Текстовый процессор MS WORD 2010 в составе пакета Microsoft Office»: учебное пособие к лабораторным работам для студентов бакалавриата сельскохозяйственных вузов. 2-е издание / Сост. А.Г. Семёнова, Е.В. Тимошкина, Е.С. Третьякова. – Ижевск: ФГБОУ ВПО Ижевская ГСХА, 2013. – 48 стр. + Портал ИжГСХА <http://192.168.88.95/>.
5. «Информатика: Табличный процессор MS Excel 2010 в составе пакета Microsoft Office»: учебное пособие к лабораторным работам для студентов бакалавриата сельскохозяйственных вузов / Сост. А.Г. Семёнова, Е.В. Тимошкина. – Ижевск: ФГБОУ ВПО Ижевская ГСХА, 2013. – 36 с. + Портал ИжГСХА <http://192.168.88.95/>.
6. «Информатика: система управления базами данных Microsoft Access»: учебно-методическое пособие к лабораторным работам для студентов бакалавриата сельскохозяйственных вузов / Сост. А.Г. Семёнова, Е.В. Тимошкина. – Ижевск: ФГБОУ ВПО Ижевская ГСХА, 2013. – 48 с.
7. Пакет прикладных программ Microsoft Office [Электронный ресурс]. – Режим доступа: <http://moodle.izhgsha.ru/>.
8. «Информатика. Создание презентаций средствами MS PowerPoint»: учебное пособие для студ. бакалавриата очной-заочной формы обуч. / Сост. А.Г. Семёнова, Е.С. Третьякова. – Ижевск ФГОУ ВПО Ижевская ГСХА, 2016. – 40 с.
9. «Информатика: учебное пособие по выполнению индивидуальных заданий» для студентов бакалавриата направлений «Агроинженерия», «Технология продукции и организация общественного питания», «Теплоэнергетика и теплотехника», «Техносферная безопасность» / Сост. И.Г. Абышева, А.Г. Семенова. – Ижевск: ФГБОУ ВО Ижевская ГСХА, 2016. – 64 с.

*Учебное пособие*

## **ИНФОРМАТИКА**

**Учебное пособие для студентов бакалавриата,  
обучающихся в аграрных вузах**

Составители:

**Третьякова Екатерина Сергеевна  
Тимошкина Елена Вячеславовна**

Технический редактор А. И. Трегубова

Электронное издание.  
Гарнитура Times New Roman.  
Уч.-изд. л. 7,4.  
ФГБОУ ВО Ижевская ГСХА,  
426069, г. Ижевск, ул. Студенческая, 11

