# МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

#### «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. И. И. ПОЛЗУНОВА»



Барнаул 2019

#### УДК 004-027.21 (075.8)

**Астахова Е. В.** Теоретические основы информатики. Учебное пособие. / Е. В. Астахова; Алт. госуд. технич. ун-т им. И. И. Ползунова.— Барнаул, 2019.-176 с.

В учебном пособии конспективно изложен лекционный материал дисциплины «Теоретические основы информатики». Пособие предназначено для студентов, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

Рекомендовано заседанием кафедры «Прикладная математика» Протокол № 8 от 22.04.2019

#### Репензенты:

профессор кафедры прикладной математики АлтГТУ им. И. И. Ползунова, к.ф.-м.н. Крючкова Е. Н. доцент кафедры информационных систем в экономике АлтГТУ им. И. И. Ползунова, к.т.н. Авдеев А. С.

#### ПРЕДИСЛОВИЕ

Чтобы быть успешным современный специалист должен владеть информационной культурой, фундамент которой закладывает дисциплина «Теоретические основы информатики». Информационная культура означает не только умение элементарно работать с компьютером, но и умение целенаправленно работать с информацией, владеть различными методами ее обработки.

Информатика интегрирует знания из многих областей. Владение основными понятиями из области информатики, информационных технологий является обязательным атрибутом профессиональной пригодности человека в обществе. Информатика является дисциплиной не только практического плана, но и дисциплиной социальной, гуманитарной направленности, способствующей формированию определенного мировоззрения. В рамках дисциплины происходит развитие как универсальных, так и профессиональных компетенций. Важно понимать значение информации в развитии современного информационного общества («кто владеет информацией – владеет миром»), осознавать опасности и угрозы, связанные с информационными процессами («предупрежден, значит вооружен»). Ваша задача заключается в освоении методов, способов и средств получения, хранения, переработки информации, в получении навыков работы с компьютером как средством управления информацией.

Учебное пособие состоит из 7 тем, и предназначено для формирования целостного представления об информатике, о возможностях, способах и целях использования информационных систем, программных средств и технологий. В конце каждой темы приведены вопросы для самоконтроля, которые могут быть полезны при подготовке к защите лабораторного сопровождения темы. Разделы с дополнительной информацией отмечены \*.

#### Тема 1 ВВЕДЕНИЕ В ИНФОРМАТИКУ

#### 1 ВВЕДЕНИЕ

## 1.1 Информатика, как естественнонаучная дисциплина

Термин «Informatique» (информатика) образован путем слияния слов information (информация) и automatique (автоматика) — "информационная автоматика или автоматизированная переработка информации". В англоязычных странах этому термину соответствует синоним computer science (наука о компьютерной технике).

Информатика — это область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров и их взаимодействием со средой применения.

В самом общем смысле под информатикой понимают фундаментальную естественную науку, изучающую процессы передачи, накопления и обработки информации. Составные части ядра информатики приведены на рисунке 1.

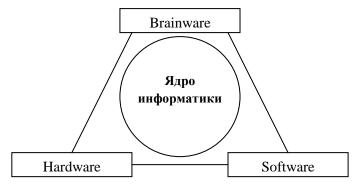


Рисунок 1 – Ядро информатики

Hardware — это устройства, имеющиеся в компьютере (технические средства).

Software — это совокупность всех программ, используемых на компьютере (программные средства).

Brainware — это то, что должен знать и уметь делать пользователь, чтобы правильно решать задачи (алгоритмические средства).

Комитет CSAB, включающий представителей Ассоциации вы-

числительной техники (ACM) и Компьютерного общества IEEE, определил четыре важнейшие области для дисциплины информатика: теория вычислений, алгоритмы и структуры данных, методология программирования и языков, компьютерные элементы и архитектура. Задачи информатики приведены на рисунке 2.



Периоды в развитии вычислительной техники (BT) и элементная база поколений приведены на рисунках 3-4.

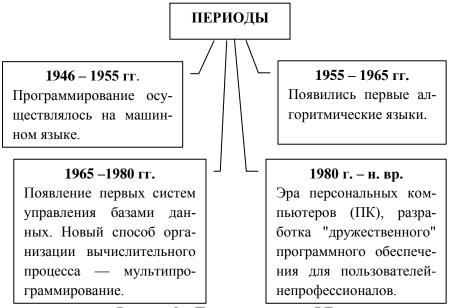


Рисунок 3 – Периоды развития ВТ

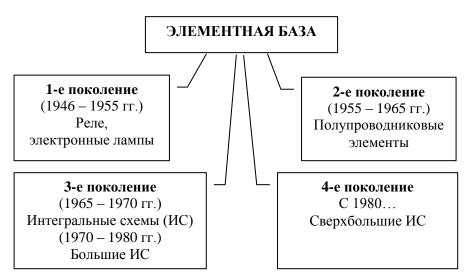


Рисунок 4 – Элементная база поколений ВТ

## \*1.3 ОБЩАЯ ХАРАКТЕРИСТИКА ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

Информационные коммуникации — это пути и процессы, обеспечивающие передачу сообщений от источника информации к ее потребителю.

Информационное взаимодействие — это взаимодействие между людьми посредством передачи данных, в результате которого происходят изменения в ощущениях, мнениях, представлениях, знаниях.

**Система** — это совокупность элементов, находящихся в отношениях и связях между собой и образующих определенную целостность.

 ${\tt Информационная}$  система — это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Информационная система — это организационно упорядоченная совокупность документов и информационных технологий, реализующих информационные процессы.

Под *информационными процессами* понимаются любые действия, выполняемые над информацией. При работе с информацией всегда имеется ее источник и потребитель. Фазы информационного процесса представлены на рисунке 5.



Рисунок 5 – Фазы, составляющие информационный процесс

Информационный процесс может проходить только в системе, в которой присутствует человек, например, процессы познания, обучения, управления.

Система создается для описания каких-то процессов. Если процесс информационный, то система обязательно должна включать людей, например, система образования, система экономики, правовая система. Цели управления системой или в системе представлены на рисунке 6.

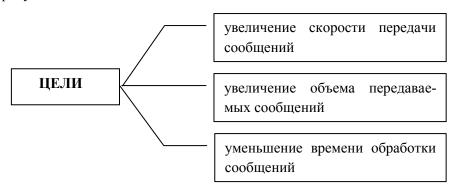


Рисунок 6 – Цели управления системой

#### \*1.4 Информационный рынок

Информационный рынок — это комплексная структура, включающая в себя правовой статус субъектов, вступающих в отношения при оказании информационных услуг и их использовании.

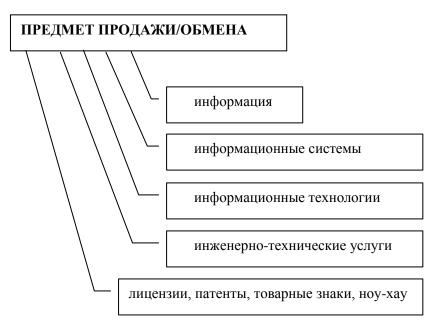


Рисунок 7 – Предмет продажи или обмена на информационном рынке

Отличительные особенности информационного продукта от других товаров:

- информация может быть использована многократно;
- информация морально устаревает со временем по мере снижения актуальности предоставляемых ею знаний;
  - информация рассчитана на разных потребителей;
- производство информации намного дороже последующего тиражирования.

Основные участники рынка информационных услуг приведены на рисунке 8.

#### УЧАСТНИКИ РЫНКА ИНФОРМАЦИОННЫХ УСЛУГ

Продавцы информации

Пользователи информации

#### Производители информации

(организации, добывающие, публикующие, обрабатывающие информацию)

Рисунок 8 – Основные участники рынка информационных услуг

#### 1.5 Информация: понятие, свойства

Термин «информация» происходит от латинского informatio, что означает разъяснение, осведомление, изложение. Информация есть отражение реального мира с помощью сведений (сообщений).

 ${
m {\it Mh}} {
m {\it dopmatus}}$  — это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.

Сообщение — это реальная форма представления информации в виде речи, текста, изображения, цифровых данных, графиков, таблиц и т. п.

**Данные** — это сигналы, образы, рассматриваемые безотно-сительно к их смыслу.

**Знание** — это продукт накопления, систематизации данных, их осмысление с целью получения информации, обеспечивающей достижение некоторой цели.

Данные могут рассматриваться как признаки или записанные наблюдения, которые по каким-то причинам не используются, а только хранятся. Если появляется возможность использовать эти данные для уменьшения неопределенности о чем-либо, то они превращаются в информацию. Таким образом, информацией являются используемые данные. Основные свойства информации приведены на рисунке 9.



Рисунок 9 – Основные свойства информации

С точки зрения потребителя качество информации оценивается такими свойствами как:

*Репрезентативность*. Обеспечивается отбором информации в целях адекватного отражения свойств объекта.

Codepжательность. Определяется смысловой емкостью информации. Это отношение количества семантической информации ( $\mathbf{I}_{\mathbf{C}}$ ) в сообщении к объему обрабатываемых данных ( $\mathbf{V}_{\mathbf{D}}$ ):

$$C = I_C / V_D$$

**Пример 1.** Объем обработанных данных составил 4 Гбайта. Содержательность 1000 Мбайт семантической информации:

$$C = I_C / V_D = 1000 / (4*1024) = 0,24.$$

**Пример 2.** Коэффициент содержательности 800 Мбайт семантической информации равен 0,15. Объем обработанных данных:

$$V_D = I_C / C = 800 / 0,15 / 1024 = 5,2$$
 Гбайт.

**Пример 3.** Объем обработанных данных составил 4 Гбайта. Коэффициент содержательности равен 0,8. Количество семантической информации:  $I_C = CV_D = 0.8*4 = 3.2$  Гбайта.

Коэффициент информативности. Это отношение количества синтаксической информации (по Шеннону) к объему данных

$$Y = I / V_D$$
.

Достаточность (полнота). Означает, что информация содержит минимальный, но достаточный для принятия правильного реше-

ния набор показателей.

*Доступность*. Обеспечивается процедурами получения и преобразования информации.

*Актуальность*. Оценивается степенью ценности информации в момент ее использования.

*Своевременность*. Означает поступление информации не позже заранее назначенного момента времени.

*Точность*. Определяется степенью близости получаемой информации к реальному состоянию объекта, процесса, явления и т. п.

*Достоверность*. Отражает реально существующие объекты с необходимой точностью.

*Устойчивость*. Означает способность информации реагировать на изменения исходных данных без нарушения необходимой точности.

#### 1.6 КОДИРОВАНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ

Код ASCII (American Standard Code for Information Interchange) используется для внутреннего представления символьной информации в редакторе Блокнот операционной системы Windows'хх, а также для кодирования текстовых файлов в Интернете.

Таблица кодов содержит 16 столбцов и 16 строк. Каждая строка и столбец пронумерованы в шестнадцатеричной системе счисления цифрами от 0 до F. Код символа складывается из номера столбца и номера строки, в которых располагается символ. ASCII-таблица делится на две части:

столбцы с номерами от 0 до 7 составляют стандарт кода — неизменяемую часть;

столбцы с номерами от 8 до F являются расширением кода и используются,в частности, для кодирования символов национальных алфавитов.

Кодировка русского языка Windows-1251 была введена компанией Microsoft для компьютеров, работающих на платформе Windows.

Кодировка *КОИ-8* (код обмена информацией восьмизначный) распространена в компьютерных сетях на территории России и в российском секторе Интернета.

*UNICODE* — универсальная 16-разрядная кодировка, позволяющая закодировать 65536 (216) различных символов.

#### 2 ИЗМЕРЕНИЕ КОЛИЧЕСТВА ИНФОРМАЦИИ

#### 2.1 МЕРЫ И ЕДИНИЦЫ ИЗМЕРЕНИЯ ИНФОРМАЦИИ

**Адекватность информации** — это уровень соответствия создаваемого с помощью полученной информации образа реальному объекту, процессу, явлению.

**Количество информации** — числовая величина, характеризующая информацию по разнообразию, сложности, упорядоченности, определенности и т.п.

**Мера** — непрерывная действительная неотрицательная функция, определенная на множестве событий.

**Вероятность** — это количественная характеристика одного из исходов некоторого опыта, известная до его проведения. Измеряется в пределах от 0 до 1.

Так как информация в различных предметных областях может пониматься и интерпретироваться по-разному, то различаются и способы определения ее количества. За основную единицу измерения количества информации принят 1 бит. Для оценки состояния системы, которая может принимать одно из п возможных состояний, используется мера информации (события).

#### 2.1.1 Мера Р. Хартли

Формула Хартли:

#### $H = k \log_a N$ ,

где k — коэффициент пропорциональности, а — основание системы меры. Формула Хартли не работает в случае различных состояний системы.

Хартли предложил оценивать количество информации в зависимости от мощности алфавита:

#### $I = L \log_a h$ ,

где L — длина сообщения, h — мощность исходного алфавита (количество символов в алфавите).

Формула Хартли для двоичной системы счисления:

#### $H = log_2N$ (бит).

**Пример 1**. Чтобы узнать положение точки в системе из 4-х клеток необходимо задать два вопроса ("Левая или правая клетка?").

В итоге, узнав положение точки, мы увеличиваем суммарную информацию о системе на 2 бита ( $I = log_2 4$ ).

**Пример 2.** Определить количество информации, содержащейся в двоичной цифре. Имеем исходный алфавит  $\{0,1\}$ . Следовательно, h=2. Длина сообщения L=1 (одна двоичная цифра). Тогда  $I=1*\log_2 2=1$  бит. Это аналитическое определение бита. На одну двоичную цифру приходится 1 бит информации.

#### 2.1.2 Мера К. Шеннона

Шеннон предложил связать количество информации в сообщении с вероятностью получения этого сообщения. Сумма вероятностей всех исходов опыта равна 1.

Формула Шеннона дает оценку информации независимо от ее смысла:

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

где n — число состояний системы;  $p_i$  — вероятность (относительная частота) перехода системы в i-е состояние (сумма всех  $p_i$  равна 1).

Исходы называются *равновероятными*, если они могут совершиться с равной долей вероятности. Для равновероятных состояний справедливы отношения:

$$p_i = 1/n, I = log_2n.$$

В общем случае выполняется неравенство:

$$I = -\sum_{i=1}^{n} p_i \log_2 p_i \le \log_2 n$$

Сообщение о наступлении события с меньшей вероятностью несет в себе больше информации, чем сообщение о наступлении события с большей вероятностью.

Сообщение о наступлении достоверно наступающего события несет в себе нулевую информацию.

Пусть по результатам некоторого опыта получено п сообщений с вероятностью  $p_i$ . Тогда количество информации в i-том сообщении определяется по формуле:

$$I = -\log_2 p_i$$
.

**Пример.** Определить количество информации в сообщении о результатах сдачи экзамена студентом, если известны вероятности получения оценок:

```
«отлично» — p(«5») = 0,1; «хорошо» — p(«4») = 0,2; «удовлетворительно» — p(«3») = 0,3; «неудовлетворительно» — p(«2») = 0,4. Количество информации в каждом сообщении будет равно: I(«5») = -\log_2 0,1 = 3,32; I(«4») = -\log_2 0,2 = 2,32; I(«3») = -\log_2 0,3 = 1,74; I(«2») = -\log_2 0,4 = 1,32.
```

Если у опыта два равновероятных исхода, то  $I_{cp}=-(0.5*\log_20.5+0.5*\log_20.5)=1$ . Это аналитическое определение *бита* по Шеннону.

#### \*2.1.3 Термодинамическая мера

В реальной жизни всегда присутствует некоторая степень неопределенности. Л. Бриллюэн охарактеризовал информацию как отрицательную энтропию. Так как энтропия является мерой неупорядоченности, то информация может быть определена как мера упорядоченности материальных систем.

Пусть потребитель имеет предварительные сведения о системе  $\alpha$  до получения информации. Н ( $\beta$ ) — мера его неосведомленности о системе (или мера неопределенности состояния системы).  $I_{\beta}(\alpha)$  — дополнительная информация после получения сообщения  $\beta$ , уменьшившая неопределенность состояния системы  $H_{\beta}(\alpha)$ .

Формула для определения количества информации о системе, полученной в сообщении  $\beta$ :

$$I_{\beta}(\alpha) = H(\alpha) - H_{\beta}(\alpha)$$

Другими словами, количество информации измеряется изменением неопределенности состояния системы.

Энтропия системы  $H(\alpha)$  может рассматриваться как мера недостающей информации.

Информационно-термодинамический подход связывает величину энтропии системы с недостатком информации о внутренней структуре. Число состояний определяет степень неполноты сведений о системе.

**Пример.** Пусть дана термодинамическая система (процесс) S.  $H_0$ ,  $H_1$  — термодинамические энтропии системы S в начальном (равновесном) и конечном состояниях термодинамического процесса. Тогда термодинамическая мера информации (*негэнтропия*) определяется формулой:  $H(H_0,H_1)=H_0-H_1$ . Уменьшение  $H(H_0,H_1)$  говорит о приближении термодинамической системы S к состоянию статического равновесия, а увеличение — об удалении.

Пусть до начала процесса можно дать  $p_1$  равновероятных ответов на вопрос о состоянии термодинамической системы, а после окончания процесса —  $p_2$  ответов. Изменение информации определяется выражением:

#### $\Delta I = k \ln(p_1 / p_2) = k (\ln p_1 - \ln p_2).$

Если  $p_1 > p_2 \ (\Delta I > 0),$  то сведения о системе стали более определенными.

Если  $p_1 < p_2$  ( $\Delta I < 0$ ) — менее определенными.

Величина  $\Delta I$  может быть интерпретирована как количество информации, необходимой для перехода от одного уровня организации системы к другому (при  $\Delta I > 0$  — более высокому, а при  $\Delta I > 0$  — более низкому уровню организации).

#### 2.1.4. Семантический подход

Для оценки количества информации с позиций целесообразности используется формула:

$$\mathbf{I} = \log(\mathbf{p}_1 / \mathbf{p}_2),$$

где  $p_1$  — вероятность достижения цели *после* получения сообщения,  $p_2$  — вероятность достижения цели  $\partial o$  получения сообщения.

**Пример.** Определить количество информации, содержащейся в шпаргалке, если известно, что вероятность сдачи экзамена без шпаргалки равна 0,2. Использование шпаргалки увеличивает вероятность сдачи до 0,8. Следовательно,  $I = \log_2(0.8/0.2) = \log_2 4 = 2$ .

Пусть шпаргалка содержала неверную информацию, вследствие чего вероятность сдачи экзамена уменьшилась до 0,1. Тогда  $I = \log_2(0,1/0,2) = \log_20,5 = -1$ . Отрицательный знак результата говорит о дезинформации.

# \*2.2 Положения комбинаторики, используемые в измерении информации

**Комбинаторика** — раздел дискретной математики, изучающий способы формирования подмножеств из элементов исходного множества.

Из конечного счетного множества элементов мощности h можно сформировать комбинации элементов (рисунок 1).



Рисунок 1 – Комбинации конечного счетного множества элементов

1. Сочетания — группировка элементов исходного множества в подмножества одинаковой мощности так, что элементы в них отличаются составом, а порядок элементов произвольный.

Число сочетаний из h элементов по L без повторений:

$$C(_{h}^{L}) = h!/(L!(h - L)!).$$

**Пример**. Пусть  $\{a,b,c\}$  — исходное множество мощности h=3. По правилу сочетаний можно получить следующие подмножества мощности 2:  $\{a,b\}$ ,  $\{a,c\}$ ,  $\{b,c\}$ . Множества типа  $\{a,b\}$ ,  $\{b,a\}$  являются идентичными и не формируются.

Число сочетаний из трех элементов по два без повторений:

$$C(^{2}_{3}) = 3!/(2!(3-2)!) = 3.$$

2. Перестановки — группировка элементов исходного множества в подмножества одинаковой мощности так, что элементы в них отличаются только порядком.

Число перестановок из h элементов без повторений:

$$\Pi(\mathbf{h}) = \mathbf{h}!$$

**Пример**. Пусть  $\{a,b,c\}$  — исходное множество мощности h=3. По правилу перестановок можно получить следующие подмножества:  $\{a,b,c\}, \{a,c,b\}, \{b,c,a\}, \{b,a,c\}, \{c,a,b\}, \{c,b,a\}.$ 

Число перестановок из трех элементов по три:

$$\Pi(3) = 3! = 6.$$

3. *Размещения* — группировка элементов исходного множества в подмножества одинаковой мощности так, что элементы в них отличаются и составом и порядком.

Число размещений из h элементов по L без повторений:

$$P(_{h}^{L}) = h!/(h - L)!$$

Пример 1. Пусть {a,b,c} — исходное множество мощности h=3. По правилу размещений можно получить следующие подмножества мощности 2: {a,b}, {a,c}, {b,c}, {b,a}, {c,a}, {c,b}.

Число размещений из трех элементов по два без повторений:

$$P(^{2}_{3}) = 3!/(3-2)! = 6.$$

Число размещений из h элементов по L с повторениями:

$$P^{\Pi}(_{h}^{L}) = h^{L}$$
.

**Пример 2.** Пусть {0,1} — исходное множество мощности h=2. По правилу размещений можно получить следующие подмножества мощности 3:  $\{0,0,0\}$ ,  $\{0,0,1\}$ ,  $\{0,1,0\}$ ,  $\{1,0,0\}$ ,  $\{1,1,0\}$ ,  $\{1,0,1\}$ ,  $\{0,1,1\}$ ,  $\{1,1,1\}.$ 

Число размещений из двух элементов по три с повторениями:  $P^{\pi}(^{3}_{2})=2^{3}=8.$ 

$$P^{\Pi}(^{3}_{2}) = 2^{3} = 8.$$

#### Вопросы для самоконтроля

- 1. Дать понятие информатики как науки.
- 2. В чем заключаются основные задачи информатики?
- 3. Какая элементная база использовалась в различных поколениях ЭВМ?
- 4. Что понимается под информационными процессами?
- 5. Дать понятия информации, сообщения, данных.
- 6. Перечислить свойства информации с позиций ее потребления.
- 7. Кто предложил связать количество информации в сообщении с вероятностью получения этого сообщения?
- 8. Дать аналитическое определение бита по Шеннону.
- 9. Примеры кодировок текстовой информации.
- 10. .Что может служить предметом продажи или обмена на информационном рынке?

#### Тема 2 ЛОГИЧЕСКИЕ И АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЭВМ

#### 1 ЛОГИЧЕСКИЕ ОСНОВЫ ЭВМ

#### 1.1 ОСНОВНЫЕ ПОНЯТИЯ АЛГЕБРЫ ЛОГИКИ

Алгебра логики (булева алгебра) занимается исчислением высказываний и применяется для описания работы дискретных устройств, к которым относятся и ЭВМ.

**Утверждение** — основная единица, неделимая с точки зрения отражения смысла информации.

**Высказывание** — это повествовательное утверждение, про которое можно однозначно сказать, истинно оно или ложно.

**Логические связки** — это слова и словосочетания «не», «и», «или», «если ..., то», «тогда и только тогда».

**Таблица истинности** — это таблица всех значений некоторой логической функции.

Структура формальной математической системы приведена на рисунке 1.



Рисунок 1 – Множества математической системы

Сигналы в логических схемах могут быть представлены бинарными символами  $\{0,1\}$ . Элементы бинарной алгебры  $B = \{0,1\}$  называются логическими нулем и единицей.

Логические связки позволяют из элементарных высказываний строить новые составные высказывания. Истинность составного высказывания зависит от истинности или ложности элементарных высказываний.

**Пример.** Элементарные высказывания «Иванов — студент», «Иванов — футболист». Связка «И» дает составное высказывание:

«Иванов — студент, хорошо играющий в футбол».

Базовые логические операции булевой алгебры приведены на рисунке 2. Приоритет операций: 1 — HE, 2 — И, 3 — ИЛИ.

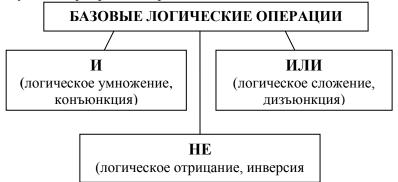


Рисунок 2 — Базовые логические операции

Таблица истинности базовых операций булевой алгебры

Переменные		Логические операции		
X	Y	$X \wedge Y$	X∨Y	$\neg X$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Кроме базовых операций существуют такие как:

- 1. Импликация  $(X \rightarrow Y) \equiv (\neg X \lor Y)$
- 2. Эквивалентность  $(X \leftrightarrow Y) \equiv (X \land Y \lor \neg X \land \neg Y)$

Логические функции эквивалентны, если совпадают их таблицы истинности.

#### 1.2 ФОРМАЛИЗАЦИЯ ВЫСКАЗЫВАНИЙ

**Выполнимая** - формула, при некоторых значениях переменных принимающая значение «истина», а при некоторых - значение «ложь».

**Тавтология** - всегда истинная формула. **Противоречие** - всегда ложная формула.

Формализация высказывания заключается в замене его логической формулой, составленной из логических переменных и символов логических операций.

**Пример**. Иванов, Петров и Сидоров утверждают, что видели виновника ДТП. Иванов говорит, что это были белые «Жигули», Пет-

ров — что черная «Волга», Сидоров — что «Москвич», но не белый. Как определить истинный цвет и марку автомобиля?

Составим простые высказывания: A = «машина — белая», B = «машина — жигули», Z = «машина — черная», U = «машина — волга», V = «машина — москвич»

На основе высказываний запишем логические выражения для каждого очевидца

Иванов: Петров: Сидоров:  $A \wedge B$  С $\wedge D$  ¬ $A \wedge E$ 

Пусть в каждом из этих логических выражений одна из переменных принимает значение «истина». Тогда будут истинны и дизъюнкции вида:

$$A \lor B = 1$$
  $C \lor D = 1$   $\neg A \lor E = 1$ 

По определению конъюнкции получим следующее выражение  $(A \lor B) \land (C \lor D) \land (\neg A \lor E) = 1$ 

Два высказывания о цвете и марке машины не могут быть одновременно истинными, значит, автомобиль был черными жигулями.

#### 1.3 ЛОГИЧЕСКИЕ СХЕМЫ

**Синтез логической схемы** заключается в соединении известных элементарных логических схем в новую схему.

**Анализ логической схемы** состоит в установлении ее выходных значений по значениям логических входов.

**Логический элемент компьютера** — это часть электронной логической схемы, которая реализует базовую (элементарную) логическую операцию (функцию).

На основе набора {И, ИЛИ, HE} можно представить *любую* логическую функцию.

Для реализации схем **И, ИЛИ** используют графическое изображение базового вентиля (рисунок 3).

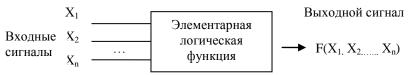


Рисунок 3 – Базовый вентиль

Для реализации элементарных схем **И-НЕ, ИЛИ-НЕ** используют графическое изображение *инвертора* на выходе (рисунок 4).



Рисунок 4 – Элементарный вентиль

#### 2 АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЭВМ

#### 2.1 Системы счисления

**Система счисления** — это способ изображения чисел с помощью конечного множества символов.

Разряд (позиция) - место для цифры в числе.

Разрядность - количество цифр в числе.

Основание системы счисления — это количество различных цифр, используемых для изображения числа в позиционной системе счисления.

В зависимости от способа изображения чисел системы счисления (c/c) делятся на позиционные и непозиционные.

В позиционной системе счисления количественное значение каждой цифры зависит от ее места (позиции) в числе.

В непозиционной системе счисления цифры не меняют своего количественного значения при изменении их расположения в числе.

Число, равное основанию системы счисления, записывается только в двух позициях данной системы.

Разряды числа нумеруются справа налево.

Пример.

Разряды	210	5 4 3 2 1 0
Число	3 7 5 <sub>10</sub>	$1\ 1\ 0\ 1\ 0\ 1_2$

Каждому разряду соответствует степень основания.

Пример.

Степень	$10^2 10^1 10^0$	$2^5 2^4 2^3 2^2 2^1 2^0$
Разряды	2 1 0	5 4 3 2 1 0
Число	3 7 510	1 1 0 1 0 1 <sub>2</sub>

Значения цифр в алфавите системы счисления лежат в пределах от 0 до P - 1, где P — основание системы счисления.

#### Пример.

Десятичная система счисления.

Основание: 10. Алфавит: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

Двоичная система счисления.

Основание: 2. Алфавит {0, 1}.

Шестнадцатеричная система счисления.

Основание: 16. Алфавит: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9,

$$A = 10$$
,  $B = 11$ ,  $C = 12$ ,  $D = 13$ ,  $E = 14$ ,  $F = 15$ .

Для системы счисления с основанием P максимальная цифра в одном разряде равна P - 1.

Разделение числа на целую и дробную части имеет смысл только позиционных системах.

В общем случае запись любого смешанного числа в системе счисления с основанием P будет представлять собой ряд вида:

$$N_p = \sum_{n=k-1}^{-m} a_n P^n, P = a_n + 1 - o$$
снование ,  $0 \le a_n \le P, A = \{a_1, a_2, \mathbf{K}, a_n\} - a$ лфавит

**Пример.** Запись смешанного числа в десятичной системе счисления:  $-851,12 = -(8*10^2 + 5*10^1 + 1*10^0 + 1*10^{-1} + 2*10^{-2}) =$ = -(800 + 50 + 1 + 0,1 + 0,02)

Наиболее эффективными являются системы с основанием, кратным 2, т.е. 2, 4, 8, 16. В силу специфики построения схем ЭВМ в современных машинах применяется *шестнадцатеричная* система счисления. Любая позиционная система счисления дает более компактную запись числа по сравнению с двоичной.

#### 2.2 АРИФМЕТИЧЕСКИЕ ДЕЙСТВИЯ В ПОЗИЦИОННЫХ СИСТЕМАХ СЧИСЛЕНИЯ

# Двоичная система счисления 1+0=1 1-1=0 1\*0=0 0+1=1 1-0=1 0\*1=0 0+0=0 0-0=0 0\*0=0 $1+1=\underline{1}0$ $0-1=\underline{2}1$ 1\*1=1 (2-1=1)

#### Восьмеричная система счисления

$$7 + 0 = 7$$
  $7 - 7 = 0$   $7 * 0 = 0$   $0 + 7 = 7$   $0 + 0 = 0$   $0 + 0 = 0$   $0 - 0 = 0$   $0 - 7 = \boxed{8}1$   $7 * 1 = 1$   $(8 - 7 = 1)$ 

#### Шестнадцатеричная система счисления

$$F + 0 = F 
0 + F = F 
0 + 0 = 0 
F + 1 = 10 
F - F = 0 
F - F = 0 
0 * F = 0 
0 * F = 0 
0 * O = 0 
0 * O = 0 
1 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
0 * F = F 
1 * F = F 
(16 - F = 1)$$

2, 8, 16 — единицы заема из соседнего слева разряда.

<u>1</u> — единица переноса в соседний слева разряд.

## **Пример 1.** Умножение в двоичной системе счисления

Последовательность вычислений:

- 1) Первое слагаемое переписываем первый множитель.
- 2) Второе слагаемое переписываем первый множитель со сдвигом влево на 1 разряд.
- 3) Третье слагаемое переписываем первый множитель со сдвигом влево на 2 разряда.

**Пример 2.** Умножение в восьмеричной системе счисления

$$\begin{array}{r}
345, 2_8 \\
 & 5, 4_8 \\
\hline
 & 16250_8 \\
 & + 21722_8 \\
\hline
 & 2354, 7_8
\end{array}$$

Последовательность вычислений первого слагаемого:

1) 
$$4*2 = 8$$

$$2)$$
  $8/8 = 10$  (0 пишем, 1 в уме)

3) 
$$4*5 = 20$$

4) 
$$20/8 = 24$$
 (4+1 = 5 пишем, 2 в уме)

5) 
$$4*3 = 12$$

6) 
$$12/8=14$$
 (4+2 = 6 пишем, 1 пишем)

# **Пример 3.** Умножение в шестнадцатеричной системе счисления

Последовательность вычислений:

- 1) 4\*8=32
- 2) 32/16=20
- 3) 5\*8=40
- 4) 40/16=28
- 5) E=14\*8=112
- 6) 112/16=70
- 7) 72100 = 72A0

Правила арифметики во всех позиционных системах счисления одинаковы.

В восьмеричной системе счисления каждая цифра соответствует трем двоичным разрядам — триаде.

В шестнадцатеричной системе счисления каждая цифра соответствует четырем двоичным разрядам — *mempade*.

#### 2.3 Сложение по модулю

Сложение по модулю  $(\oplus)$  — это операция арифметического сложения, при котором единица переноса в старший разряд отбрасывается.

При выполнении операции сложения по модулю необходимо конкретизировать, о каком модуле идет речь.

**Пример 1.** Сложить по модулю 2 числа  $101_2$  и  $111_2$ .

 Разряд единиц:
  $1 \oplus 1 = 0$  

 Разряд десятков:
  $0 \oplus 1 = 1$  

 Разряд сотен:
  $1 \oplus 1 = 0$ 

Результат: 010.

Чтобы подчеркнуть, что в сложении участвовали трехразрядные слагаемые, в результате оставляют три цифры.

**Пример 2.** Сложить по модулю 10 числа  $69_{10}$  и  $142_{10}$ .

 Разряд единиц:
  $9 \oplus 2 = 1$  

 Разряд десятков:
  $6 \oplus 4 = 0$  

 Разряд сотен:
  $0 \oplus 1 = 1$ 

Результат: 101.

**Пример 3**. Сложить по модулю 3 числа  $69_{10}$  и  $142_{10}$ .

 Разряд единиц:
  $9 \oplus 2 = 2$  

 Разряд десятков:
  $6 \oplus 4 = 1$  

 Разряд сотен:
  $0 \oplus 1 = 1$ 

Результат: 112.

#### 3 ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ЭВМ

#### 3.1 ФОРМЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ

В вычислительных машинах применяются две формы представления чисел (рисунок 1).



Рисунок 1 – Формы представления чисел в ЭВМ

 $\mbox{\it Числа}\mbox{\it C}\Phi\mbox{\it T}$  изображаются в виде последовательности цифр с постоянным для всех чисел положением точки/запятой, отделяющей целую часть от дробной. Эта форма наиболее проста, естественна. Однако, она имеет небольшой диапазон представления чисел и поэтому не всегда приемлема при вычислениях.

Диапазон значащих чисел (N) в системе счисления с основанием Р при наличии k разрядов в целой части и m разрядов в дробной части числа (без учета знака числа) будет:

$$P^{-m} \leq N \leq P^k - P^{-m},$$

(при P = 2, k = 10 и m = 6 получим  $0.015 \le N \le 1024$ )

**Пример.** Числа 521,377; 0,000238; -102013,036, записанные в разрядную сетку  $k=5,\,m=5$ :

+00521,37700; +00000,00238; -10201,30360.

Если в результате операции получится число, выходящее за допустимый диапазон, происходит *переполнение разрядной сетки*, и дальнейшие вычисления теряют смысл. В современных ЭВМ естест-

венная форма представления используется как вспомогательная и только для целых чисел.

Числа СПТ изображаются в виде двух групп цифр. Первая группа цифр называется мантиссой, вторая — порядком. Абсолютная величина мантиссы должна быть меньше 1, а порядок — целым числом. В общем виде число в форме с плавающей запятой может быть представлено так:

$$N = \pm MP^{\pm r}$$

где M — мантисса числа (|M|<1); r — порядок числа; P — основание системы счисления.

Вещественный формат с m-разрядной мантиссой позволяет абсолютно точно представлять m-разрядные целые числа.

Любое целое число может быть без искажений преобразовано в вещественный формат

Мантисса должна быть правильной дробью, у которой первая цифра после запятой отлична от нуля. При выполнении этого требования число называется *нормализованным*.

Числа (521,377; 0,000238; -102013,036) в нормальной форме запишутся так:

$$+0.521377*10^3$$
  $+0.238*10^{-3}$   $-0.102013036*10^5$ 

Нормальная форма представления имеет огромный диапазон отображения чисел и является основной в современных ЭВМ.

Диапазон значащих чисел в системе счисления с основанием P при наличии k разрядов у мантиссы и m разрядов у порядка (без учета знаковых разрядов порядка и мантиссы) будет:

$$P^{-k}P^{-(P^m-1)} \le N \le (1-P^{-k})P^{(P^m-1)}$$

**Пример.** При  $P=2,\ k=10$  и m=6 диапазон чисел простирается примерно от  $10^{-19}$  до  $10^{19}$ .

Чем больше разрядов отводится под мантиссу, тем выше точность представления числа.

Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего, отличного от нуля числа, до наибольшего, представимого в заданном формате.

#### \*3.2 Особенности представления чисел в ЭВМ

Разрядная сетка машины имеет постоянное число разрядов.

В представлении чисел СФТ точка всегда находится перед старшим разрядом. Все числа, участвующие в вычислениях, по модулю меньше единицы.

 $\mathcal{L}$ иапазон изменения характеризуется пределами, в которых могут находиться числа:

$$|X|_{max} = 0.111...1 = 1-2^{-n}$$
  
 $|X|_{min} = 0.00...01 = 2^{-n}$ .

Числа, выходящие за диапазон изменения, не могут быть представлены в ЭВМ точно.

Ошибка представления числа зависит от величины самого числа и способа округления. При оптимальном округлении абсолютная ошибка  $|\Delta X| \leq 0.5^{*2^{-n}}$ , относительная ошибка  $2^{-(n+1)} \leq |\delta X| \leq 0.5$ 

$$|\delta X|_{min} = |\Delta X|/|X|_{max}, \; |\delta X|_{max} = |\Delta X|/|X|_{min}.$$

Если число меньше минимального, то оно воспринимается как машинный нуль.

Если число больше максимального, то оно воспринимается как машинная бесконечность.

Для чисел СПТ абсолютная ошибка равна  $|\Delta X| \leq 0.5^*2^{-m}$ , относительная  $|\delta X|_{min} \approx 2^{-(m+1)}$ , при т — большом, и  $|\delta X|_{max} = 2^{-m}$ .

Относительная ошибка для чисел СПТ не зависит от порядка числа.

При сложении чисел, представленных в форме с плавающей точкой, необходимо уравнять порядки слагаемых, сложить мантиссы и нормализовать результат.

При умножении чисел, представленных в форме с плавающей точкой, перемножают мантиссы, а значения порядков суммируют.

#### 3.3 ПРЯМОЙ, ОБРАТНЫЙ, ДОПОЛНИТЕЛЬНЫЙ КОДЫ

Положительные числа во всех кодах одинаковы.

Старший разряд регистра — знаковый (0 кодирует знак«+», 1 кодирует знак «-»).

Для выполнения действий над отрицательными числами вводят *обратный* и *дополнительный* коды.

Алгоритмы преобразования чисел в обратный и дополнительный коды приведены на рисунках 2-4.

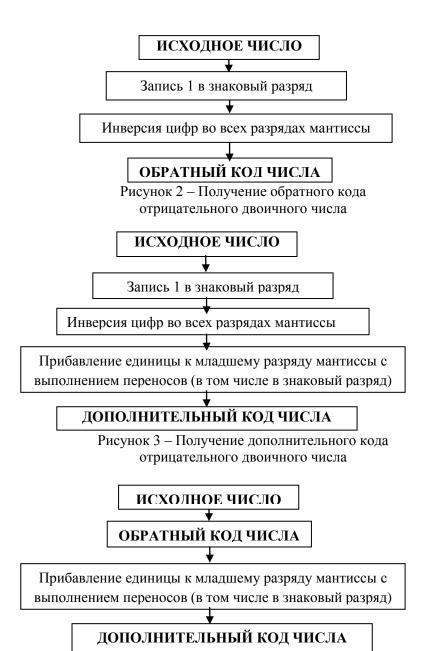


Рисунок 4 – Получение дополнительного кода отрицательного двоичного числа из обратного кода

Если числа разных знаков, то в прямом коде приходится выполнять операцию вычитания после предварительного сравнения кодов чисел. Результату приписывается знак большего числа. Эти дополнительные операции приводят к дополнительному усложнению сумматора.

При сложении чисел в *обратном коде*, складываются все n разрядов, включая знаковый. В случае возникновения *переноса* в знаковом разряде к младшему разряду *добавляется* 1, т.е. выполняется циклический перенос.

При сложении чисел в *дополнительном коде* складываются все n разрядов, включая знаковый. В случае возникновения переноса в знаковом разряде 1 отбрасывается.

При сложении в *прямом коде не делается перенос* из старшего цифрового разряда в знаковый.

Пример 1. Сложение обратных кодов чисел.

Сумма обратных кодов равна обратному коду результата.

a)  
0, 
$$00100 = [A]_{o6p} = A$$
  
+  $0, 01100 = [B]_{o6p} = B$   
0,  $10000 = [C]_{o6p} = C$ 

**6)** A = -0, 01001 1, 
$$10110 = [A]_{o6}p$$

 $\underline{B}=0,01110$  0,  $01110=[\underline{B}]_{\underline{o}\underline{o}\underline{p}}$  Выполняется циклический перенос, что равноценно суммированию с 0, 00001.

$$10,00100$$
 $0,00101 = [C]_{\text{обр}} = C$  После циклического

переноса.

B)
$$A = -0, 01100$$

$$[A]_{oбp} = 1, 10011$$

$$+$$

$$B = 0, 01001$$

$$C = -0, 00011$$

$$[B]_{oбp} = 0, 01001$$

$$1,11100 = [C]_{oбp}$$

r)
$$A = -0, 01100$$

$$1, 10011 = [A]_{oбp}$$

$$+$$

$$1, 10110 = [A]_{oбp}$$

$$C = -0, 10101.$$

$$1, 10110 = [A]_{oбp}$$

$$1, 01010 = [C]_{oбp}$$

**Пример 2.** Сложение чисел, представленных в дополнительном коде.

Сумма дополнительных кодов есть дополнительный код результата.

$$\begin{array}{l} \textbf{a)} \\ 0,\, 01001 = [A]_{\text{доп}} = [A]_{\text{пр}} \\ + \\ \underline{0,\, 00101} = [B]_{\text{доп}} = [B]_{\text{пр}} \\ 0,\, 01110 = [C]_{\text{доп}} = [C]_{\text{пр}} \end{array}$$

A = -0, 10010

C = 1, 10000

б) Нет переноса из мантиссы в знаковый разряд.

 $1,01110 = [A]_{\text{поп}}$ 

$$egin{array}{l} + \\ B = 0,01010 & 0,01010 = [B]_{non} \\ C = -0,01000 & 1,11000 = [C]_{non} \\ B) \\ A = 0,10001 = [A]_{non} = [A]_{np} \\ + \\ B = 0,11111 = [B]_{non} = [B]_{np} \\ \end{array}$$

При сложении положительных чисел в результате переноса в знаковом разряде появилась единица. Результат неверен.

Произошли переносы из младшего разряда в знаковый разряд и из знакового разряда.

$$A$$
)  $A = -0, 10001$   $1, 01111 = [A]_{доп}$   $+$   $B = -0, 11010$   $1, 00110 = [B]_{доп}$   $10, 10101 = [C]_{доп}$ 

Произошел перенос из знакового разряда. Этот перенос в сумматоре, работающем в дополнительном коде, теряется. Получилось положительное число. Результат неверен.

#### 3.4 Модифицированные коды

Переполнение разрядной сетки возникает в результате сложения чисел, когда старшие разряды результата не помещаются в отведенной для него области памяти. Переполнение разрядной сетки в результате переноса из знакового разряда удобно обнаруживать в модифицированных кодах.

В модифицированных обратном и дополнительном кодах для кода знака отводится 2 разряда. Знак «—» обозначается кодом 11, а знак «+» — кодом 00. Единица левее знаковых разрядов теряется.

Сочетания 01 и 10 служат признаком переполнения разрядной сетки.

Пример 1. Модифицированный обратный код.

a)  

$$A = 0, 10001$$
  $00, 10001 = [A]^{M}_{ofp}$   
 $B = -0, 00111$   $11, 11000 = [B]^{M}_{ofp}$   
 $C = 0,01001$ 

<u>1</u>00, 01010

В результате циклического переноса из старшего знакового разряда в младший разряд мантиссы получили положительное число 00, 01001.

**6)**

$$A = -0, 10110$$

$$[A]^{M}_{oбp} = 11, 01001$$

$$+$$

$$B = -0, 01011$$

$$[B]^{M}_{oбp} = 11, 10100$$

$$110,11101$$

С учетом циклического переноса получаем 10, 11110 в знаковом разряде переполнение.

в) 
$$A = -0, 10010 \qquad 11, 01101 = [A]^{M}_{oбp}$$
 
$$\frac{B = -0, 01011}{C = -0,11101} \qquad \frac{11, 10100 = [B]^{M}_{oбp}}{11, 00001}$$
 С учетом циклического переноса получаем 11,00010 г) 
$$A = 0, 10011 \qquad 00, 10011 = [A]^{M}_{oбp}$$
 
$$B = 0, 11001 \qquad 00, 1^{\dagger}_{001} = [B]^{M}_{oбp}$$
 01, 01100 Результат неверен.

Пример 2. Модифицированный дополнительный код.

$$A = 0, 1010$$

$$00, 1010 = [A]_{\text{доп. м}}$$
.

$$B = -0,0101$$

100, 0101 Единица, расположенная ле-

вее двух знаковых разрядов, отбрасывается, получаем: 00, 0101

$$A = -0, 1010$$

$$[A]_{\text{доп.}}^{\text{M}} = 11,0110$$

$$B = -0,0101$$

$$[B]^{M}_{\text{доп.}} = 11, 1011$$
 $111,0001$ 

С учетом потери 1 получаем 11, 0001

в)

$$A = -0, 1010$$

$$11,0110 = [A]^{M}_{\text{доп.}}$$

$$B = -0, 1010$$

$$\frac{11,0110 = [B]^{M}_{доп.}}{110,1100}$$

отбрасывания В результате левой 1 получаем 10, 1100.

Сочетание 10 в знаковых разрядах говорит о переполнении разрядной сетки.

**Γ)**A = 00, 1001
+

B = 00, 1001

C = 01, 0111

Сочетание 01 говорит о переполнении разрядной сетки.

#### Вопросы для самоконтроля

- 1. Перечислить базовые логические операции в порядке убывания приоритета.
- 2. Чем разряд числа отличается от разрядности в позиционных системах счисления?
- 3. Есть ли разница между переводом и кодированием чисел?
- 4. Сколько двоичных цифр требуется для кодирования одной восьмеричной цифры?
- 5. С каким кодом связано понятие циклического переноса?
- 6. Где в разрядной сетке находится старший разряд?
- 7. Что означает переполнение разрядной сетки?
- 8. В чем состоит модификация обратного кода?
- 9. Какие числа представимы в формате СФТ?
- 10. При сложении в каком коде единица переноса в знаковом разряде отбрасывается?

# Тема 3 ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

#### 1 УСТРОЙСТВО ЭВМ

#### 1.1 Состав и назначение основных блоков

**Компьютер** - универсальное многофункциональное электронное программно-управляемое устройство для обработки, хранения и обмена информацией с внешними объектами.

 $\Pi 3 y$  — постоянное запоминающее устройство — строится на основе установленных на материнской плате модулей (кассет); используется для хранения неизменяемой информации, например, загрузочных программ операционной системы (ОС), программ тестирования устройств  $\Pi K$ ; это энергонезависимая память.

**ОЗУ** - оперативное запоминающее устройство - предназначено для хранения и считывания информации, непосредственно участвующей в вычислительном процессе; это энергозависимая память.

Регистровая кэш-память — высокоскоростная память, является буфером между оперативной памятью (ОП) и микропроцессором.

**Шина** (магистраль) — это среда передачи сигналов, к которой может параллельно подключаться несколько компонентов вычислительной системы и через которую осуществляется обмен данными.

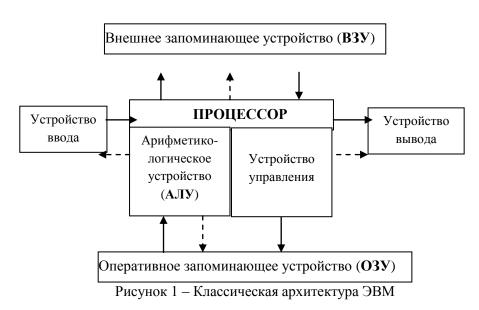
**Сумматор** — вычислительная схема; может иметь разрядность двойного машинного слова.

**Контроллер** - плата, управляющая работой периферийного устройства (дисководом, винчестером, монитором) и обеспечивающая его связь с материнской платой.

**Чипсет** — комплект микросхем с системной логикой. Может быть реализован в виде раздельных чипов, либо в виде интегральной монокристальной схемы.

Форм-фактор (типоразмер) системной платы определяет ее размеры, тип разъема питания, расположение элементов крепления, размещение разъемов различных интерфейсов.

Основы учения об архитектуре ЭВМ заложил американский математик Джон фон Нейман. Он выдвинул основополагающие принципы логического устройства ЭВМ и предложил структуру, которая применялась в ЭВМ 1-2 поколений (рисунок 1).



Типы устройств персонального компьютера приведены на рисунке 2.



Рисунок 2 – Типы устройств персонального компьютера

Процессор (центральный процессор) — функционально законченное программно-управляемое устройство обработки информации, выполненное в виде одной или нескольких интегральных схем.

*Устройство управления* формирует и передает во все блоки машины управляющие импульсы; формирует адреса ячеек памяти, используемых выполняемой операцией.

Арифметико-логическое устройство (АЛУ) предназначено для выполнения всех арифметических и логических операций над числовой или символьной информацией. АЛУ является составной частью микропроцессора.

Конструктивно ПК выполняется в виде системного блока (рисунок 3), к которому подключаются внешние устройства. Базовая аппаратная конфигурация включает системный блок, клавиатуру, монитор (дисплей). Конфигурацию можно дополнять внешними (периферийными) устройствами, к которым относятся принтер, модем, сканер, колонки и др.

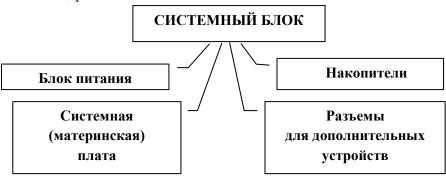


Рисунок 3 – Устройства системного блока

Для объединения функциональных устройств ПК в вычислительную систему используется *системная шина*, которая включает шины данных, адреса, инструкций (управления), питания.

Основная память предназначена для хранения и оперативного обмена информацией с блоками ПК; содержит ПЗУ (ROM) и ОЗУ (RAM). Структурно основная память состоит из ячеек емкостью 1 байт каждая. В кэш-памяти хранятся данные, которые микропроцессор получил и будет использовать в ближайшие такты своей работы. ОЗУ, ПЗУ и кэш относятся к устройствам внутренней памяти.

Состав материнской платы приведен на рисунке 4.

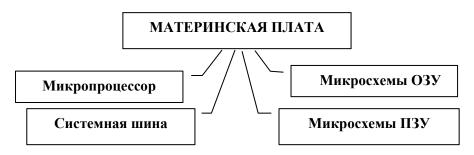


Рисунок 4 – Устройства материнской платы

*Чипсет* обеспечивает работу процессора, системной шины, интерфейсов взаимодействия с оперативной памятью и другими компонентами компьютера.

Функциональные характеристики ПК:

- 1. Производительность, тактовая частота.
- 2. Разрядность ПК и шин интерфейса.
- 3. Емкость оперативной памяти и накопителей.
- 4. Параметры накопителей.
- 5. Вид и емкость кэш-памяти.
- 6. Тип видеомонитора и видеоадаптера.
- 7. Имеющееся программное обеспечение
- 8. Тип операционной системы.
- 9. Возможность работы в вычислительной сети.
- 10. Надежность, стоимость, габариты и вес.

## \*1.2 ФУНКЦИОНИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

**Алгоритмическая универсальность** - способность вычислительной машины решать вычислительные и логические задачи из любой области человеческой деятельности.

**Точность вычислений** - мера погрешности, определяемая абсолютными и относительными ошибками.

**Скорость вычислений** определяется затратами времени на решение задач и зависит от структуры ЦВМ и быстродействия отдельных ее частей.

#### 1.2.1 Свойства вычислительных машин

Основные свойства цифровых вычислительных машин приведены на рисунке 5.



Рисунок 5 – Свойства цифровых вычислительных машин

Сущность программного управления заключается в том, что сигналы управления работой отдельных частей ЭВМ вырабатываются внутри машины в процессе вычислений. Источниками информации о требуемых типах сигналов на каждом шаге вычислений является код команды Код команды считывается из памяти машины. Программа вычислений в виде последовательности команд, закодированных в цифровой форме, записывается в память ЭВМ до начала вычислений. Поскольку команды программы внутри ЭВМ неотличимы от других цифровых кодов, то в процессе выполнения программы можно их изменить. За счет этого свойства порядок процесса вычислений в машине может изменяться в зависимости от знака результата.

## 1.2.2 Принципы функционирования ЭВМ

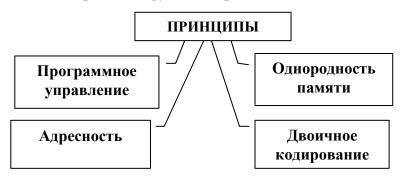


Рисунок 6 – Принципы функционирования ЭВМ (по фон Нейману)

*Принцип программного управления* обеспечивает автоматизацию процессов вычислений на ЭВМ (рисунок 7).

Принцип однородности памяти. Отсутствие принципиальной разницы между программой и данными дает возможность ЭВМ самой формировать программу в соответствии с результатами вычислений.

Принцип адресности. Структурно основная память состоит из пронумерованных ячеек. Процессору в произвольный момент доступна любая ячейка.

*Принцип двоичного кодирования* означает, что вся информация, поступающая в ЭВМ, кодируется с помощью двоичных сигналов.



Рисунок 7 – Основные положения программного управления (по фон Нейману)

#### 1.2.3 Машинные команды

Операционная часть команды — это группа разрядов в команде, предназначенная для представления кода операции машины.

Адресная часть команды — это группа разрядов в команде, в которых записываются коды адреса ячеек памяти машины, предназначенных для оперативного хранения информации.

Операнды - числа, участвующие в операции.

Состав машинных команд по видам выполняемых операций приведен на рисунке 8.

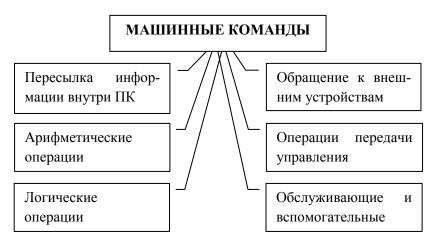


Рисунок 8 – Машинные команды по видам выполняемых операций

Операции передачи управления (ветвление программы) служат для изменения естественного порядка выполнения команд. Бывают операции безусловной и условной передачи управления. Операции безусловной передачи управления требуют выполнения команды, адрес которой указан в адресной части. Операции условной передачи управления требуют передачи управления по адресу в адресной части команды, если выполняется некоторое условие, указанное в коде операции.

Решение задач на ЭВМ реализуется программным способом, путем выполнения операций, предусмотренных алгоритмом решения задачи. По количеству адресов, записываемых в команде, команды делятся на безадресные, одно-, двух- и трехадресные.

T	иповая стр	эуктура	трехад	ресной	команды
	КОП	<b>A</b> 1	A2	A3	

КОП — код операции; А1 и А2 — адреса ячеек (регистров), где расположены соответственно первое и второе числа, участвующие в операции; А3 — адрес ячейки (регистра), куда следует поместить число, полученное в результате выполнения операции.

Типовая структура двухадресной команды КОП А1 А2

A1 — обычно адрес ячейки (регистра), где хранится первое из чисел, участвующих в операции, и куда после завершения операции должен быть записан результат; A2 — обычно адрес ячейки (регистра), где хранится второе участвующее в операции число.

Типовая структура одноадресной команды КОП А1

A1 — в зависимости от модификации команды может обозначать либо адрес ячейки (регистра), где хранится одно из чисел, участвующих в операции, либо адрес ячейки (регистра), куда следует поместить число — результат операции.

Безадресная команда содержит только код операции; информация для нее должна быть заранее помещена в определенные регистры машины. Безадресные команды могут использоваться только совместно с командами другой адресности.

## 1.3 ПРОЦЕССОРЫ

**Поток** — это последовательность команд/данных, обрабатываемая процессором.

**Радиатор** - кусок металла с относительно большой площадью поверхности (устанавливается на процессоре для его эффективного охлаждения).

Технологические нормы производства процессоров определяют допустимое расстояние между элементами на кристалле и минимально возможный их размер. Чем меньше расстояние, тем больше элементов можно разместить на единице площади кристалла.

Практически вся электрическая мощность процессора преобразуется в тепловую энергию, которую необходимо отводить. Средний ресурс процессора составляет порядка 100 тыс. часов, т.е. примерно 30 лет при ежедневной 10 часовой работе. Однако процессоры морально устаревают максимум через 2 года после своего выпуска.

В архитектуре многопроцессорных вычислительных систем используют понятие потока (рисунок 9). В гарвардской архитектуре вычислительных систем потоки команд и данных разделены, а в принстонской — нет.



Рисунок 9 – Классификация многопроцессорных систем

#### \*1.4 Типы и характеристики интерфейсов

**Интерфейс** — совокупность средств сопряжения и связи устройств ПК, обеспечивающая их эффективное взаимодействие.

Интерфейс — это аппаратное и программное обеспечение, предназначенное для сопряжения систем или частей системы.

Внутримашинный системный интерфейс — совокупность электрических линий связи (проводов), схем сопряжения с компонентами ПК, протоколов (алгоритмов) передачи сигналов.

Сопряжение систем или частей системы посредством интерфейса подразумевает действия, приведенные на рисунке 10.



Рисунок 10 – Функции сопряжения

Технические характеристики интерфейсов приведены на рисунке 11.

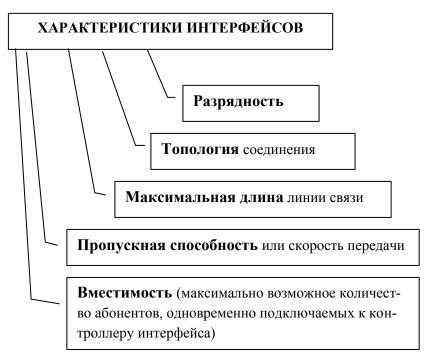


Рисунок 11 – Технические характеристики интерфейсов

По функциональному назначению выделяют:

- системные интерфейсы (интерфейсы, связывающие отдельные части компьютера как микропроцессорной системы);
  - интерфейсы периферийных устройств.

**Интерфейс SCSI.** Устройства, подключенные к шине SCSI, могут играть роль *Ведущего* (Initiator) и/или *Ведомого* (Target). Взаимодействие между устройствами осуществляется по принципу «отправитель – адресат». Отправитель инициирует запрос. Ждет ответа от адресата, после чего начинает обмен данными. Данные по шине SCSI передаются *синхронно* (с подтверждением получения пакета данных) или *асинхронно* (с подтверждением получения каждого байта).

*Инфракрасный интерфейс IrDA* позволяет передавать данные 10-битными словами: 8 бит данных, один стартовый бит в начале и один стоповый бит в конце посылки. Воздушный промежуток меж-

ду устройствами позволяет принять инфракрасную энергию только от одного источника в данный момент времени. Схема обращения устройств — это протокол обмена данными с запросами и ответами между первичным устройством и вторичным. Первичное устройство отвечает за организацию соединения и за обработку ошибок. Посланные им кадры называются управляющими. Пакеты вторичных устройств именуются ответными.

Интерфейс USB представляет собой последовательную полудуплексную (данные попеременно передаются в обоих направлениях) шину. Архитектура USB — топология «звезда». В системе должен быть корневой концентратор, к которому подключаются периферийные концентраторы, а к ним — устройства USB. Периферийные устройства могут подключаться друг к другу, образуя каскады. Физически USB представляет собой две скрученные пары проводов. По одной паре передаются данные в каждом направлении, по другой — питание (+5 В).

## 1.5 ОСНОВНЫЕ УСТРОЙСТВА

## 1.5.1 Клавиатура

**Клавиатура** — устройство ввода в ПК команд, данных и управляющих воздействий.

Блок механической клавиатуры содержит датчики клавиш, электронные схемы дешифрации и контроллер, который состоит из буферной памяти и схемы управления.

Контроллер клавиатуры осуществляет:

- опрос состояния клавиш;
- запоминание в буфер отдельных кодов клавиш на время между двумя соседними опросами клавиатуры микропроцессором;
  - преобразование кодов нажатия клавиш в коды ASCII;
  - тестирование клавиатуры при включении ПК;
  - автоматическое повторение клавишной операции.

Основной принцип работы клавиатуры заключается в сканировании переключателей клавиш. Замыканию и размыканию любого из переключателей соответствует уникальный цифровой код — scanкod. При работе с клавиатуры в буфер контроллера поступает код нажатия (0) или отпускания (1) клавиши. По наличию кода отпускания идет проверка, все ли клавиши отпущены в момент нажатия следующей клавиши, если нет, то возможно была использована комбинация клавиш (например, Ctrl+F1). Если клавиша нажата более полсекунды,

то генерируются повторные коды нажатия клавиши через регулярные интервалы времени.

### 1.5.2 Монитор

**Разрешение** — количество пикселей по вертикали и горизонтали.

**CRT** — электронно-лучевая трубка.

**LCD** — жидкокристаллический дисплей.

**Анизотропия** — зависимость свойств среды от направления — основная особенность жидких кристаллов.

**ТFT** — тонкопленочный транзистор (толщиной 0.1 — 0.01 мкм).

Видеоадаптер выполняет согласование монитора с системной шиной, вырабатывая сигналы синхронизации и видеосигналы, которые подаются на вход монитора. Видеоадаптеры снабжены оперативной памятью, в которой хранится информация об изображении в виде матрицы. Объем видеопамяти влияет на разрешение и количество воспроизводимых цветов.

Основные характеристики мониторов и проекторов, связанные с качеством изображения, приведены на рисунке 12.



Рисунок 12— Характеристики монитора, влияющие на качество изображения

Мониторы на базе электронно-лучевых трубок (ЭЛТ) по принципу действия мало чем отличаются от обычного телевизора. Испускаемый электронной пушкой пучок электронов, попадая на экран, вызывает его свечение. На внутреннюю поверхность экрана кинескопа нанесены люминофорные элементы с заданным спектральным диапазоном. Люминофорные элементы светятся под воздействием электронного потока.

Любое текстовое или графическое изображение на экране монитора состоит из множества дискретных точек люминофора, именуемых пикселями (pixel — picture element). Поэтому такие дисплеи называют растровыми. В случае цветного монитора имеются три электронных пушки с отдельными схемами управления, а на поверхность экрана нанесен люминофор трех основных цветов: R (Red, красный), G (Green, зеленый), В (Blue, синий). Эти цвета называются первичными. Путем их сложения в определенном количестве можно получить любой другой цвет. Электронные пучки, испускаемые электронной пушкой, проходят по строке, подсвечивая нужные пиксели с заданной интенсивностью.

Разрешающая способность и цветовая палитра определяются как возможностями самого монитора, так и возможностями контроллера. Качество изображения, получаемого на экране монитора, зависит от параметров электронно-лучевой трубки (рисунок 13)

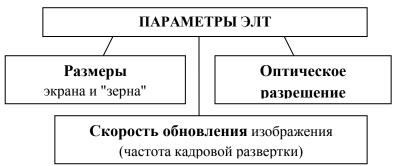


Рисунок 13 — Параметры ЭЛТ, влияющие на качество изображения В цветном мониторе разрешение соответствует одной триаде *RGB* (рисунок 14).



Рисунок 14 – Триада RGB

Время послесвечения люминофора и частота регенерации экрана должны соответствовать друг другу. Если послесвечение больше, то наблюдается размытость и удвоение кадров, если меньше, то заметно мелькание изображения. При регенерации с частотой в 60  $\Gamma$ ц. наблюдается мерцание экрана (рекомендуется частота 100  $\Gamma$ ц).

**Жидкокристаллические мониторы.** Жидкие кристаллы (ЖК) — нечто среднее между кристаллической структурой и жидкостями — представляют собой сложные органические вещества. Приложенное к ЖК-слою напряжение  $(1,5-3\ B)$  приводит к изменению ориентации жидкого кристалла. Под воздействием переменного поля, группы молекул кристалла скручиваются или выпрямляются, что приводит к рассеиванию лучей света. Разбиение экрана на определенное количество пикселей происходит по специальной матрице. Матрица состоит из двух слоев с взаимно перпендикулярными проводниками.

Структу	ра электрооптической ячейки				
	Стеклянная пластинка				
Про	оводящий слой — электрод				
Проклад	ка из непроводящего материала				
Ж	идкий кристалл 5 – 50мкм				
Проклад	ка из непроводящего материала				
Про	рводящий слой — электрод				
	Стеклянная пластинка				

Для передачи цветного изображения на ЖК-панель наносят цветофильтр (RGB). Размер и форма цветовой ячейки определяется размером и формой электрооптической ячейки. На поверхность дисплея наносятся 2 пленки: одна рассеивает свет, вторая компенсирует эффект запаздывания света. Запаздывание возникает в случае, если свет проходит через жидкий кристалл с наклоном. Световой поток должен попадать в ЖК-ячейки строго под 90° равномерно по всей поверхности.

Плазменные мониморы. На стеклянные пластины нанесены отдельно вертикальные и горизонтальные проводники. Пространство между панелями заполнено инертным газом. Когда к проводникам прикладывается определенное напряжение, газ начинает светиться. Между пластинами с проводниками размещена вспомогательная пластина с круглыми отверстиями для локализации свечения. Пикселем является воображаемая точка на пересечении проводников.

## 1.5.3 Манипуляторы

**Мышь** — особого рода манипулятор, позволяющий оптимизировать работу с большой категорией компьютерных программ, исключить непроизводительное частое повторное нажатие некоторых клавиш.

**Микки** — единица измерения перемещения мыши (1 дюйм = 200 микки).

Разрешающая способность мыши — минимальное перемещение, которое может интерпретироваться ее воспринимающими механизмами; указывается обычно числом точек на дюйм.

**Трекбол** – манипулятор, шарик которого приводится в движение ладонью руки; не требует гладкой поверхности, устанавливается стационарно, в основном в портативных  $\Pi K$ 

**Пенмаус** – является аналогом шариковой ручки, на конце которой вместо пишущего узла установлен узел, регистрирующий величину перемещения.

**Джойстики** - манипуляторы рычажно-нажимного типа, используются для компьютерных игр.

Первую компьютерную мышь создал Дуглас Энджельбарт в 1963 году в Стэндфордском исследовательском центре.

**Принцип действия механической мыши.** На внутренних «колесах» мыши имеются токопроводящие отметки. К колесам примыкают проводящие «реснички» Их взаимодействие в процессе вращения колес шариком мыши позволяет определить направление и скорость движения курсора.

Принцип работы оптической мыши. Фотодатчики установлены на нижней поверхности корпуса мыши. С помощью светодиода и системы линз под мышью подсвечивается участок поверхности. Отраженный от поверхности свет собирается другой линзой. Процессор обработки изображений делает снимки поверхности под мышью с высокой частотой и обрабатывает их. Последовательность снимков представляет собой квадратную матрицу из пикселей разной яркости.

Принцип работы оптико-механической мыши (рисунок 15). Утяжеленный шарик с резиновым покрытием катается по плоской поверхности и вращает два перпендикулярно расположенных валика, сообщая движение в декартовой системе координат. На конце каждого валика расположено колесо с мелкими отверстиями по окружности. Это колесо вращается между светодиодом и приемникомфототранзистором. Луч света проходит через «спицы» колеса, вра-

щающегося с разной скоростью. Порядок, в котором освещаются фоточувствительные элементы одной оси, определяет направление перемещения мыши. Частота приходящих от фоточувствительных элементов импульсов определяет скорость мыши. Информация о длительности световых импульсов преобразуется в электронные сигналы и позволяет определить скорость перемещения и положение курсора на экране.

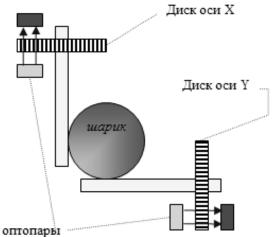


Рисунок 15 – Устройство оптико-механической мыши

Принцип работы беспроводной мыши. Связь с компьютером осуществляется через инфракрасный порт, поэтому необходим прямой «визуальный» контакт между устройством и приемопередатчиком, что, ограничивает свободу перемещения. Если используется радиоканал, то снимается ограничение на прямую видимость устройств. Однако при этом возможно влияние помех.

## \*1.6 ПЕРИФЕРИЙНОЕ ОБОРУДОВАНИЕ

К устройствам ввода графической информации относятся: сканеры, графические планшеты (дигитайзеры). Сканеры бывают: планшетные, ручные, барабанные, сканеры форм, штрих-сканеры. Исходный материал вводится в графическом виде, затем обрабатывается программами распознавания образов.

Планшетные сканеры предназначены для ввода графической информации с прозрачного или непрозрачного листового материала. Луч света, отраженный от поверхности материала или прошедший

через него фиксируется *приборами с зарядовой связью* (ПЗС). Обычно элементы ПЗС делают в форме линейки, располагаемой по ширине исходного материала. Основные потребительские параметры планшетных сканеров приведены на рисунке 16.

## ПАРАМЕТРЫ СКАНЕРА

## Разрешающая способность

(зависит от плотности размещения ПЗС на линейке, от точности позиционирования при сканировании)

## Динамический диапазон

(определяется логарифмом отношения яркости наиболее светлых участков изображения к яркости наиболее темных участков)

#### Производительность

(определяется продолжительностью сканирования листа бумаги)

Рисунок 16 – Потребительские параметры сканера

Ручные сканеры. Принцип действия ручных сканеров соответствует планшетным сканерам. Разница заключается в том, что протягивание линейки ПЗС выполняется вручную. Равномерность и точность сканирования при этом неудовлетворительная.

Барабанные сканеры. Исходный материал закрепляется на цилиндрической поверхности барабана, вращающегося с высокой скоростью. Наивысшее разрешение (2400 – 5000 dpi) обеспечивается благодаря применению фотоэлектронных умножителей. Используются для сканирования исходных изображений высокого качества.

Сканеры форм. Используются для сканирования форм, заполненных от руки (при переписи населения, обработки выборов и т.п.). Качество сканирования низкое, главное — быстродействие.

*Штрих-сканеры*. Это разновидность ручных сканеров используется для чтения и ввода информации со штрих-кода.

Графические планшеты (дигитайзеры) предназначены для ввода художественной графической информации. Принцип действия основан на фиксации перемещения специального пера относительно планшета.

## 2 УСТРОЙСТВА ХРАНЕНИЯ ИНФОРМАЦИИ

#### 2.1 Классификация устройств

- 1. *Ленточные накопители* (библиотеки на магнитных лентах, накопители на сменных картриджах стримеры).
  - 2. Дисковые накопители.
    - 2.1 CD накопители (CD-ROM, CD-R, WORM, DVD).
  - 2.2 Накопители на гибких магнитных дисках (3,5", 5,25", накопители Zip).
  - 2.3 *Накопители на жестких магнитных дисках* (классические винчестеры, RAID системы).
    - 2.4 Магнитооптические накопители.
    - 3. Накопители на твердотельной памяти.
      - 3.1.*Модули flash-памяти*.
      - 3.2. Модули оперативной памяти.
      - 3.3. Модули графической памяти.

### 2.2 ХАРАКТЕРИСТИКА УСТРОЙСТВ ПАМЯТИ

**Память ЭВМ** — совокупность устройств, служащих для запоминания, хранения и выдачи информации.

**Запоминающие устройства** – это отдельные устройства, относящиеся к памяти  ${\tt ЭВМ}$ .

Производительность и вычислительные возможности ЭВМ в значительной степени определяются составом и характеристиками ее запоминающих устройств (ЗУ).

Основные операции в памяти приведены на рисунке 1.



Рисунок 1 – Операции обращения к памяти

## Характеристики ЗУ

*Емкость памяти* определяется максимальным количеством данных, которые могут в ней храниться.

*Быстродействие памяти* определяется продолжительностью операции обращения к памяти, т. е. временем, затрачиваемым на такие операции как:

- поиск нужной единицы информации в памяти;
- считывание;
- поиск места в памяти, предназначенного для хранения данной единицы информации;
  - запись в память.

При обработке информации осуществляется взаимодействие процессора и оперативной памяти (рисунок 2).



Рисунок 2 – Обработка информации в компьютере

Запоминающее устройство любого типа состоит из запоминающего массива, хранящего информацию, и блоков, служащих для поиска в массиве, записи и считывания.

## \*2.3 НОСИТЕЛИ ИНФОРМАЦИИ

**Плотность Записи** — это объем информации, который можно надежно разместить на единице площади поверхности носителя.

**Кластер** (ячейка размещения данных) — это наименьший участок диска, с которым оперирует операционная система при записи файла. Кластер состоит из одного или нескольких секторов.

**Цилиндр** — это все дорожки, одновременно находящиеся под головками чтения записи.

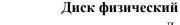
Раздел - независимые логические части диска.

**Скорость передачи данных** — максимальная скорость, с которой данные пересылаются от носителя информации в оперативную память компьютера.

Коэффициент ошибок — это оценка вероятности искажения информационного бита при его считывании с загрязненного или поцарапанного участка диска.

## 2.3.1 Гибкие магнитные диски

Поверхность гибкого магнитного диска разбивается на дорожки, начиная с внешнего края (нулевая дорожка). Дорожки разбиваются на секторы (рисунок 3).



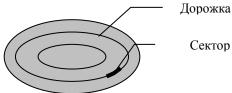


Рисунок 3 – Физическая структура диска

Плотность записи бывает:

- *радиальная* (поперечная); измеряется числом дорожек, размещенных на кольце диска шириной 1 дюйм (tpi).
- *линейная* (продольная); измеряется количеством бит, которое можно записать на дорожке единичной длины (dpi).

#### 2.3.2 Жесткие магнитные лиски

Жесткий диск (винчестер) представляет собой пакет, состоящий из нескольких дисков, стороны дисков нумеруются, начиная с верхней стороны верхнего диска. Дорожки жесткого диска разбиваются на секторы, которым присваиваются номера, начиная с 0. Нулевой сектор резервируется для идентификации записываемой информации.

Основа диска покрывается тонким слоем магнитного вещества. Оксидный рабочий слой представляет полимерное покрытие с наполнителем из окиси железа. Рабочий слой на основе *тонких пленок* имеет меньшую толщину и более прочен. На поверхность магнитного рабочего слоя наносится высокопрочное защитное покрытие.

Винчестеры служат в компьютерах основными устройствами массовой памяти и, как правило, содержат несколько магнитных дисков (пластин). Количество магнитных головок чтения/записи равно

количеству рабочих поверхностей. Все головки смонтированы на общем подвижном каркасе и перемещаются одновременно. Чем ближе располагаются головки к поверхности рабочего слоя, тем ближе друг к другу располагаются домены, тем выше емкость диска.

Заполнение винчестеров гелием позволяет: установить в них до семи пластин (дисков); уменьшить мощность электромоторчиков шпинделя; увеличить точность позиционирования головок; снизить уровень шума и тепловыделение.

Плотность гелия в 7 раз ниже плотности воздуха, что позволяет уменьшить силу трения, действующую на магнитные пластины внутри винчестера, повысить точность позиционирования головок и пластин.

PMR — технология перпендикулярной магнитной записи. Плотность записи с помощью обычной перпендикулярной технологии PMR приближается снизу к 1 Тбит/дюйм $^2$  (на 2016 г.)

Пластины жестких дисков с перпендикулярной записью изготавливаются из сплава кобальта, хрома и платины, частицы которого имеют диаметр 8 нм и длину 16 нм. Для записи одного бита головке необходимо намагнитить около 20 таких частиц.

SMR (черепичная запись с перекрытием) даёт возможность преодолеть рубеж плотности в 1 Тбит/дюйм<sup>2</sup>. Вместе с технологией TDMR плотность записи на пластине составит 1,4 Тбит/дюйм<sup>2</sup>, а с технологией HAMR — до 2-3 Тбит/дюйм<sup>2</sup>.

HGST представила первый в мире коммерческий жёсткий диск с гелием внутри в 2013 году. Seagate начала продажи своих «гелиевых» накопителей ёмкостью 10 Тбайт весной 2015 года. (3dnews.ru)

ТDMR (двухмерная магнитная запись) — технология, которая должна способствовать росту производительности жёстких дисков за счет увеличения плотности записи, делая дорожки уже, а ячейки меньше. Вместо одной головки для чтения используется несколько (3 − 5) головок со сдвигом. Головки могут располагаться друг за другом или фронтом с частичным перекрытием флангов. Это позволяет фильтровать помехи и считывать данные с более узкой дорожки.

Жесткие диски Seagate Enterprise Performance 10К. Скорость вращения шпинделя — 10 тысяч оборотов в минуту; несколько считывающих устройств на каждой головке; плотность записи свыше 1 Тбит/дюйм<sup>2</sup>. (3dnews.ru)

*HAMR* — технология термомагнитной записи. Используются головки, которые локально нагревают пластину до 450 С, используя лазер с длиной волны 810 нм и мощностью 20 мВт.

Плотность записи у винчестеров на базе HAMR составляет примерно 2  ${\sf Тбит/дюйм}^2$  (3dnews.ru)

#### 2.3.3 Накопители СD

 ${ t CD-ROM}$  — это компакт-диск (CD), предназначенный только для хранения в цифровом виде записанной на него информации и считывания ее.

**CD-R** — диск однократной записи.

 $\mathtt{CD-RW}$  — перезаписываемый диск.

Информация, хранимая на CD-ROM (рисунок 4) практически не подвержена разрушению электромагнитными полями и стиранию в результате естественного старения материала носителя. Толщина диска составляет 1.2 мм, диаметр — 120 мм. Прозрачный поликарбонатный диск покрыт с одной стороны тонким металлическим отражающим слоем и защитной пленкой прозрачного лака. Двоичный нуль представляется на диске как в виде углубления, так и в виде основной поверхности, а двоичная единица — в виде границы между ними.

Зона ввода — область в форме кольца шириной 4 мм, ближайшего к центру диска. С нее начинается считывание информации. Содержит оглавление, адреса записей, число заголовков, суммарное время записи, название диска.

Зона данных (файловая система) — область в форме кольца шириной 33 мм.

Зона вывода содержит метку конца диска.

Хранение данных на CD-ROM как и на магнитных дисках, организуется в двоичной форме в виде древовидной файловой структуры. При кодировании 1 байта информации на диске записывается 14 бит плюс 3 бита. Базовая информационная единица —  $\kappa a \partial p$ . Кадры на диске образуют секторы и блоки.

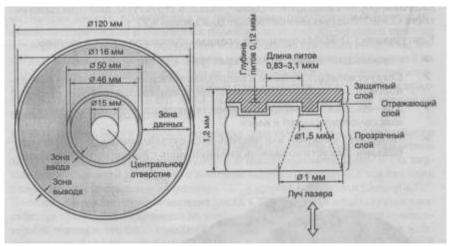


Рисунок 4 – Геометрические характеристики компакт-диска

#### Этапы изготовления СD

- 1. *Подготовительный этап*. Формирование информационного файла для последующей записи на носитель.
- 2. Запись на носитель данных с помощью лазерного луча в виде последовательности расположенных по спирали углублений (штрихов). Глубина каждого штриха (pit) равна 0.12 мкм, ширина 0.8 мкм. Штрихи расположены вдоль спиральной дорожки; расстояние между соседними витками составляет 1.6 мкм, что соответствует плотности 16 000 витков/дюйм (625 витков/мм). Длина штрихов вдоль дорожки записи 0.8 3.3 мкм.
- 3. Проявление фоторезистивного слоя и металлизация диска. Изготовленный по такой технологии диск называется мастердиском. Для тиражирования компакт-дисков с мастер-диска методом гальванопластики снимается несколько рабочих копий. Рабочие копии покрыты более прочным металлическим слоем, чем мастер-диск и могут использоваться в качестве матриц для тиражирования до 10 тыс. СD с помощью каждой матрицы. Тиражирование дисков осуществляется методом горячей прессовки. После прессовки информационная сторона диска подвергается вакуумной металлизации слоем алюминия, а затем покрывается слоем лака. Диски, выполненные методом горячей прессовки, обеспечивают до 10 000 циклов безошибочного считывания данных.

Накопители CD-WORM и CD-R позволяют единожды записать информацию на диск и многократно ее считывать. Диск покрыт термочувствительным слоем красителя. При записи информации луч лазера разогревает слой золота и слой красящего вещества. В результате химической реакции облучаемый участок на поверхности диска изменяет свой цвет и, следовательно, свои отражательные свойства. Считывающий лазер накопителя CD-ROM воспринимает эти участки как псевдоуглубления с меньшим уровнем интенсивности отражаемого света.

#### 2.3.4 Накопители DVD

В накопителях используется более узкий луч лазера, чем в приводах CD-ROM, поэтому толщина защитного слоя диска была снижена до 0,6 мм. Так как общая толщина диска должна остаться неизменной (1,2 мм), под предохранительный слой поместили укрепляющий, на котором также стали записывать информацию. Так появились двухслойные диски. При считывании лазерным лучом информации, записанной на первом слое в глубине диска, луч беспрепятственно проходит через полупрозрачную пленку, образующую второй слой. По окончании считывания информации с первого слоя, луч фокусируется в плоскости наружного полупрозрачного слоя для дальнейшего считывания данных.

#### 2.3.5 Магнитооптические накопители

Слои одностороннего диска (сверху вниз):

- защитный;
- диэлектрический
- магнитооптический;
- диэлектрический;
- отражающий;
- подложка.

На стеклопластиковую подложку для отражения лазерного луча наносится алюминиевое (золотое) покрытие. Диэлектрические слои (прозрачный полимер) окружающие магнитооптический слой, защищают диск от перегрева, повышают чувствительность при записи и отражающую способность при считывании информации. Магнитооптический слой создается на основе порошка из сплава кобальта, железа и тербия. Верхний защитный слой из прозрачного полимера предохраняет рабочую поверхность от механических повреждений.

#### 2.3.6 Накопители на магнитной ленте

**Стример** - устройство для резервного копирования больших объемов информации.

Tехнология DDS. Магнитная лента проходит через блок головок, установленных на вращающемся барабане. Записывается столько наклонных дорожек, сколько магнитных головок в блоке.

*Технология DLT*. Магнитная лента, разделенная на параллельные горизонтальные дорожки, проходит через неподвижную магниторезистивную головку со скоростью  $2.5-3.7\,\mathrm{m/c}$ , что обеспечивает малый износ магнитного слоя ленты. Расчетный срок службы ленты —  $500\,000\,$  перемоток.

## 2.3.7 Накопители на твердотельной памяти

В накопителях на твердотельной памяти носителями информации являются полупроводниковые элементы. Эти элементы конструктивно выполнены в виде интегральных CMOS-микросхем и называются *твердотельной памятью*.

Микросхемы Flash-памяти являются энергонезависимыми и перезаписываемыми. Flash-память устойчива к неблагоприятным внешним воздействиям, имеет сравнительно малое среднее время доступа к данным, низкое энергопотребление, высокую емкость, небольшой вес и размеры. Накопители на Flash-памяти сохраняют работоспособность при температурах от -40 до + 85 °C, выдерживают ударные ускорения до  $1000 \, \mathrm{g}$ .

## Вопросы для самоконтроля

- 1. Какие устройства входят в архитектуру ЭВМ?
- 2. Для чего предназначена кэш-память?
- 3. Перечислить известные типы интерфейсов.
- 4. Как работает контроллер клавиатуры?
- 5. Перечислить параметры мониторов, влияющие на качество изображения?
- 6. Как формируется цветное изображение в мониторе?
- 7. Перечислить потребительские параметры сканера?
- 8. Классы устройств хранения информации.
- 9. На какие классы делятся манипуляторы типа мышь?
- 10. Отличие физической структуры диска от логической?

## Тема 4 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

## 1 АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

#### 1.1 ПОНЯТИЕ АЛГОРИТМА

**Алгоритм** — это некоторый конечный набор операций, рассчитанных на определенного исполнителя, в результате выполнения которых через определенное число шагов может быть достигнута поставленная цель или решена задача.

Вспомогательные или подчиненные алгоритмы — это готовые алгоритмы, целиком включаемые в состав разрабатываемого алгоритма.

Слово «алгоритм» использовалось в математике для обозначения правил выполнения четырех арифметических действий: сложения, вычитания, умножения и деления. В настоящее время его применяют во многих областях человеческой деятельности, например, говорят об алгоритме управления производственным процессом, алгоритме поиска пути в лабиринте и т. п.

Значение слова «алгоритм» близко по смыслу к значению слов «указание» или «предписание». Алгоритм указывает последовательность действий по переработке исходных данных в результаты.

Основные особенности алгоритма приведены на рисунке 1.



Рисунок 1 – Основные свойства алгоритма

*Массовость*. Алгоритм имеет некоторое число входных величин — аргументов, задаваемых до начала работы. Один и тот же алго-

ритм применим для работы на разных наборах исходных данных.

Дискретность. Алгоритм представлен в виде конечной последовательности шагов, т. е. имеет дискретную структуру. Выполнение каждого очередного шага начинается после завершения предыдущего.

*Конечность*. Выполнение алгоритма заканчивается после выполнения конечного числа шагов.

Определенность (детерминированность). Каждый шаг алгоритма должен быть четко и недвусмысленно определен, т.е. не должен допускать произвольной трактовки исполнителем. При повторениях алгоритма для одних и тех же исходных данных всегда получается одинаковый результат.

Эффективность. Алгоритм должен быть эффективным, т. е. действия исполнителя на каждом шаге можно выполнить точно и за разумное конечное время.

*Результативность*. Алгоритм должен завершаться определенными результатами.

Алгоритм сначала формулируется в общих чертах, а затем уточняется путем детализации более крупных действий через более мелкие. Это метод *пошаговой детализации* или метод *«верху вниз»*. При построении алгоритма по принципу «сверху вниз» задача разбивается на более простые подзадачи.

#### \*1.2 ЗАПИСЬ АЛГОРИТМА

**Команды** — отдельные указания исполнителю на каждом шаге алгоритма.

**Подпрограммы** — вспомогательные алгоритмы, записанные на языках программирования.

Существует несколько способов записи алгоритма (рисунок 2).



Рисунок 2 – Способы записи алгоритма

Если алгоритм предназначен для исполнителя-человека, то для его записи может использоваться *естественный язык*, лишь бы запись отражала все основные особенности алгоритма. Команды алгоритма нумеруют, чтобы можно было на них ссылаться.

**Пример**. Алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел:

- 1. Если числа равны, то взять первое число в качестве ответа и закончить исполнение алгоритма, иначе перейти к п. 2.
  - 2. Определить большее из двух чисел.
- 3. Заменить большее число на разность большего и меньшего чисел.
  - 4. Перейти к п. 1.

Команды такого алгоритма выполняются в естественной последовательности, если не оговорено противного.

*Блок-схемы* представляют алгоритм в наглядной графической форме. Команды алгоритма помещаются внутрь блоков, соединенных стрелками, показывающими очередность выполнения команд алгоритма. Приняты определенные стандарты графических изображений блоков. Для записи команды внутри блоков используется естественный язык с элементами математической символики.

Наглядность быстро теряется при изображении большого алгоритма, так как схема получается плохо обозримой.

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. В псевдокоде, так же как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда.

На практике в качестве исполнителей алгоритмов используются ЭВМ. Следовательно, язык для записи алгоритма должен быть формализован. Такой язык принято называть языком программирования, а запись алгоритма на этом языке — программой.

В языках программирования высокого уровня принята символическая форма записи, близкая к общепринятой математической. Выражение на языке программирования задает правило вычисления некоторого значения. Структурные компоненты выражения приведены на рисунке 3.

Выражение называют арифметическим, если его значением является число. Арифметическое выражение записывается в строку.

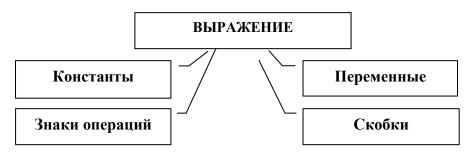


Рисунок 3 – Компоненты выражения

Действия в арифметическом выражении выполняются слева направо с соблюдением приоритета операций (рисунок 4). При одинаковом приоритете, выполняется операция, стоящая левее.



Рисунок 4 – Приоритет операций в арифметическом выражении (по убыванию)

## \*1.3 СТРУКТУРЫ АЛГОРИТМОВ

Алгоритмы можно представлять как некоторые жесткие структуры, состоящие из отдельных базовых элементов.

Простая команда является элементарной структурной единицей любого алгоритма. Она обозначает один элементарный шаг переработки или отображения информации. Переработка информации состоит в изменении значений некоторых величин, с которыми работает

алгоритм. Величины делятся на *постоянные*, значения которых остаются неизменными в процессе исполнения алгоритма, и *переменные*.

С величиной связано *имя*, используемое для обозначения. В качестве имени используется *идентификатор*. Простая команда на языке схем алгоритма изображается в виде функционального блока, имеющего один вход и один выход (рисунок 5).



Рисунок 5 – Функциональный блок простой команды

Составные команды образуются из простых команд и команды проверки условий. Команда *следования* образуется из последовательности команд.

С помощью команды ветвления (рисунок 6) осуществляется выбор одного из двух возможных действий в зависимости от условия.

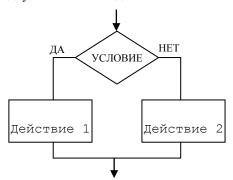


Рисунок 6 – Функциональные блоки команды ветвления (развилка)

Если условие соблюдено, то выполняется действие 1, в противном случае — действие 2. Существует команда ветвления в сокращенной форме (рисунок 7).

Команда повторяемых команд. Составная команда цикла (команда повторения) содержит условие, которое используется для определения количества повторений.

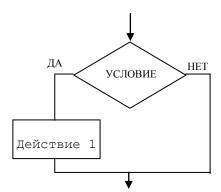


Рисунок 7 – Функциональные блоки команды ветвления (коррекция)

Команда повторения с предусловием (рисунок 8). Исполнение команды состоит в том, что сначала проверяется условие, и если оно соблюдено, то выполняется действие. После этого снова проверяется условие. Выполнение цикла завершается, когда условие перестает соблюдаться.

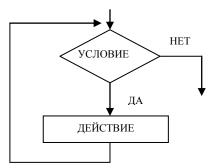


Рисунок 8 – Функциональные блоки команды повторения (с предусловием)

Команда повторения с постусловием (рисунок 9). Условие проверяется после выполнения команды. Повторное выполнение команды происходит, если условие не соблюдено. Действие, выполняемое в цикле, должно влиять на условие.

Любой алгоритм может быть построен с использованием только базовых конструкций: *следования, развилки и цикла*.

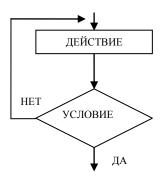


Рисунок 9 – Функциональные блоки команды повторения (с постусловием)

Построение алгоритма может происходить двумя путями:

- 1. Базовые элементы могут соединяться в последовательность, образуя конструкцию следования.
- 2. Одна базовая конструкция может вкладываться в другую конструкцию, так как внутри составных команд могут быть использованы другие составные команды.

#### \*1.4 Структуры данных

**Простые переменные** описывают структуры, состоящие из одного элемента.

**Массив** — структура, состоящая из множества элементов, упорядоченных в соответствии со значениями индексов.

**Очередь** — структура данных, организованная по принципу «первым пришел — первым ушел».

**Стек** — структура данных, организованная по принципу «последним пришел — первым ушел».

**Строка** — структура данных, описывающая последовательность символов из некоторого алфавита.

Задачи, решаемые на ЭВМ – это математические модели процессов или явлений реальной жизни, в которых отражены наиболее существенные связи между реальными объектами. Модели объектов вместе с присущими им связями образуют *структуры данных*.

Каждая *простая переменная* характеризуется одним значением

Массив как структура данных характеризуется тем, что с помощью индексов обеспечивается прямой доступ к любому элементу массива. В зависимости от числа индексов различают массивы одно-

мерные, двумерные и т. д.

Обработка элементов *очереди* ведется последовательно один за другим. Элемент, который первым попал в очередь, первым и обрабатывается и при этом покидает очередь. Добавление новых элементов производится в конец очереди.

При записи в *стек* очередной элемент заносится в его вершину, а остальные элементы продвигаются вниз без изменения порядка. При выборке из стека выбирается элемент из его вершины, а все остальные элементы без изменения порядка сдвигаются вверх, так что в вершину попадает элемент, поступивший в стек предпоследним.

## 2 СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

#### 2.1 КЛАССИФИКАЦИЯ ПРОГРАММНЫХ ПРОДУКТОВ

**Операционная система** — совокупность программных средств, которая:

- управляет аппаратной частью компьютера;
- осуществляет взаимодействие блоков компьютера;
- управляет выполнением программ;
- организует хранение информации во внешней памяти;
- поддерживает интерфейс пользователя.

Программные продукты по сфере использования делятся на три больших класса (рисунок 1). В задачи системного ПО входят:

- обеспечение надежной и эффективной работы компьютера и вычислительной сети;
- создание операционной среды функционирования других программ;
- проведение диагностики и профилактики аппаратуры компьютера и вычислительных сетей;
- выполнение вспомогательных технологических операций (копирование, архивирование, восстановление файлов и т. п.).
- Системное ПО подразделяется на базовое и сервисное.
- *Базовое программное обеспечение* представляет собой минимальный набор программных средств, обеспечивающих работу компьютера. К базовому ПО относятся операционные системы и операционные оболочки.

#### КЛАССЫ ПРОГРАММНЫХ ПРОДУКТОВ

# **Системное** программное обеспечение

(совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ)

# Прикладное программное обеспечение

(комплекс взаимосвязанных программ для решения задач в конкретной предметной области)

## Инструментарий технологии программирования

(совокупность программ и программных комплексов для разработки, отладки и внедрения создаваемых программных продуктов)

Рисунок 1 – Классификация программных продуктов

Сервисное программное обеспечение включает программы и программные комплексы, которые расширяют возможности базового ПО и организуют более удобную среду работы пользователя. В состав сервисного ПО могут входить программы:

- диагностики работоспособности компьютера;
- обслуживания сети;
- обслуживания дисков и архивирования данных;
- антивирусные и др.

Прикладные программы связаны с ОС и могут эксплуатироваться на компьютерах с аналогичной системной средой. Современная ОС должна реализовывать мультипрограммную обработку, виртуальную память, поддерживать многооконный интерфейс.

Уровни программного обеспечения вычислительной системы представляют собой пирамиду, где следующий уровень опирается на программное обеспечение предшествующих уровней (рисунок 2).

Утилиты относятся к служебному программному обеспечению. Драйверы входят в системное программное обеспечение. Программы для решения конкретных задач относятся к прикладному программному обеспечению. Базовая система ввода/вывода (BIOS) относится к базовому программному обеспечению.



Рисунок 2 – Уровни программного обеспечения

#### 2.2 ФАЙЛОВАЯ СИСТЕМА ДИСКА

Файл — именованная область на диске, где хранится логически связанная совокупность данных.

Кластер - минимальная единица пространства диска, отводимого файлу.

Каталог - это, с одной стороны, группа файлов, объединенных пользователем исходя из некоторых соображений, а с другой стороны — это файл специального вида, содержащий системную информацию о составляющих его файлах.

Права доступа к файлу — набор операций для каждого пользователя, которые он может применить к данному файлу.

Блок - наименьшая единица данных, которой внешнее устройство обменивается с оперативной памятью.

FAT - это раздел файловой системы в ОС Windows, содержащий имена файлов и папок, сведения об их размещении на носителе информации.

Файловая система является частью ОС и предназначена для обеспечения удобного интерфейса при работе пользователя с хранящимися на диске данными (рисунок 3).

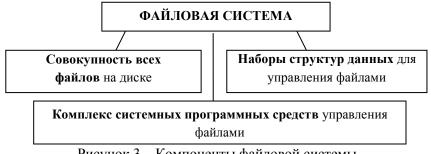


Рисунок 3 – Компоненты файловой системы

Файлы на диске записываются в свободные кластеры, поэтому фрагменты одного и того же файла могут находиться в разных местах диска. Производительность системы выше, если фрагменты файла занимают подряд идущие кластеры. Сведения о номерах кластеров хранятся в таблице размещения файлов — FAT. Файл состоит из физических записей — блоков. В каталоге содержится список файлов и устанавливается соответствие между файлами и их атрибутами. В разных файловых системах в качестве атрибутов используются разные характеристики, например:

- информация о разрешенном доступе;
- владелец/создатель файла;
- признак «только для чтения», «скрытый файл», «системный файл», «архивный файл», «временный»;
  - время создания, последнего доступа (изменения);
  - размер файла.

ОС распознает тип (формат) файла по его расширению.

Тип файла	Расширение
Исполняемые файлы	exe, com, bat
Текстовые файлы	txt, doc, rtf
Электронные таблицы	xls
Презентации	ppt
Графические файлы	gif, bmp, jpg, jpeg, tif
Звуковые файлы	wav, midi, mp3, wms
Видеофайлы	avi, mpeg
Веб-страницы	htm, html
Базы данных	dbf, mdb
Архивы данных	arj, rar, zip

#### Файловая система обеспечивает:

- возможность доступа к конкретному файлу;
- поиск свободного места при записи нового файла;
- определение схемы записи информации, содержащейся в файлах, на физический диск.

#### \*2.3 Система прерываний

**Прерывание** означает прекращение выполнения текущей команды для обработки некоторого события, после чего осуществляется возврат в выполнению прерванной программы.

**Время реакции** — это время между появлением сигнала запроса на прерывание и началом выполнения программы обработчика прерываний.

**Глубина прерывания** определяется максимальным числом программ, которые могут прерывать друг друга.

Система прерываний — основной механизм функционирования операционной системы. Запросам на прерывание присваивается номер (тип прерывания), по которому определяется адрес обработчика прерывания — специальной программы. Переход к одному из 256 обработчиков прерывания осуществляется с помощью таблицы векторов прерываний. При вызове прерывания процессор оставляет свою работу, загружает адрес программы обработки прерывания и передает ей управление. По окончании работы программы прерывания управление передается основной программе, которая была прервана.

Аппаратные прерывания инициируются аппаратурой (принтер, клавиатура). Логические прерывания возникают при нештатной ситуации (деление на 0, переполнение регистров). Программные прерывания возникают, когда одна программа хочет получить сервис со стороны другой программы.

Типы прерываний приведены на рисунке 6.



Рисунок 6 – Типы прерываний

# 3 ОПЕРАЦИОННЫЕ СИСТЕМЫ СЕМЕЙСТВА WINDOWS

## 3.1 ИСТОРИЯ СОЗДАНИЯ ГРАФИЧЕСКОЙ ОПЕРАЦИОННОЙ СРЕДЫ

**Интерфейс** — совокупность средств и правил, которые обеспечивают взаимодействие устройств, программ и человека.

Программная совместимость — способность операционной системы исполнять программные продукты, созданные в другой операционной системе.

**Окно** — прямоугольная область на экране монитора, в которой отображаются приложение, документ, сообщение.

**Приложение** — комплекс взаимосвязанных программ для создания, модификации и управления объектами определенного типа.

**Задача** — приложение, работающее в текущем сеансе и потребляющее ресурсы компьютера: оперативную память, процессорное время, доступ к внешним устройствам.

Операционная система Windows ориентирована на организацию удобной среды работы пользователя на персональном компьютере. Windows прошла путь от графической оболочки операционной системы MS DOS в первых версиях до полноценной операционной системы.

*В 1981 г.* компания Microsoft приступила к работе над графической средой для персональных компьютеров.

**Апрель 1983** г. Выпуск программы *Interface Manager*, которая после серии доработок не была принята.

*Июнь 1985* г. *Windows 1.0*. Этот вариант не нашел поддержки.

**Октябрь 1987** г. Windows 2.0. Разработан табличный процессор Microsoft Excel и Word 1.0.

**Май 1990** г. Windows 3.0 — графическая оболочка операционной системы MS DOS; полноценный графический интерфейс, поддержка режима многозадачности. Переход на более удобный способ управления компьютером с помощью мыши.

**Апрель 1992** г. Windows 3.1. Начиная с этой версии, реализована заложенная в процессоре 80386 и выше возможность организации виртуальной памяти.

*Сентябрь 1995 г. Microsoft Windows 95* — первая графическая операционная система для компьютерной платформы IBM PC.

1998 г. Windows 98 более производительна, так как файловая система FAT 32 оперирует кластерами меньших размеров (по сравнению с FAT 16), что позволяет рационально использовать диски.

**Февраль 2000 г.** Windows 2000 поддерживает компьютеры с 8, 16 и даже 32 процессорами. Увеличилась надежность и масштабируемость, возросла легкость использования, расширена поддержка аппаратного обеспечения. В Windows 2000 включено программное обеспечение для поддержки универсальной последовательной шины USB, технологии Plug-and-Play.

Октябрь 2001 г. Windows XP. В основе системы Windows XP Professional лежит проверенный код Windows NT® и Windows 2000, характеризуемый 32-разрядной вычислительной архитектурой и полностью защищенной моделью памяти. В ОС Windows XP имеются все необходимые сведения о самой операционной системе и компьютере, на котором она установлена.

Центр справки и поддержки позволяет быстро получить:

- инструкции по выполнению конкретных действий;
- статьи на интересующие темы;
- советы по устранению неполадок.

**2007** г. Windows Vista — это полноценная 64-битная система. Многие существенные изменения в ОС Windows Vista не затрагивают собственно ядро системы, а касаются усовершенствований оболочки, работы в сети и графической модели нового поколения. ОС Windows Vista содержит множество усовершенствований в области процессов и потоков, включая использование счетчика циклов центрального процессора для более равномерного выделения ресурсов.

B Windows Vista появилась новая технология — SuperFetch ("СуперВыборка"). SuperFetch определяет, какие приложения и компоненты системы наиболее часто используются пользователем и выполняет предзагрузку этих программ в оперативную память.

**2009** г. Windows 7 — операционная система семейства Windows NT, следующая за Windows Vista. В ОС Windows 7 были внесены существенные изменения, позволившие повысить скорость загрузки и производительность операционной системы, ее защищенность и стабильность, а также снизить энергопотребление. Windows 7 содержит новые средства устранения неполадок, восстановления процессов, диагностики зависаний сети без уведомления пользователя.

Windows является *многозадачной*, т.е. способна «одновременно» выполнять несколько программ. Один микропроцессор может выполнять инструкции только одной программы. Операционная система настолько быстро реагирует на потребности программ, что создается впечатление одновременности их работы.

В среде Windows система самостоятельно создает и изменяет файлы конфигурации, распознает конкретное техническое устройство и производит его автонастройку. Это технология *Plug and Play* — «включай и работай». Дополнительные устройства подключаются специальными программами-мастерами. Если устройство не поддерживает эту технологию, программы-мастера запрашивают дополнительную информацию в удобной для пользователя форме.

*Многопоточность* операционной системы означает, что работающие программы (процессы) могут разделяться на несколько частей, самостоятельно претендующих на процессорное время.

Проблема нехватки оперативной памяти решается в среде Windows с помощью *виртуальной* (реально не существующей) памяти. Виртуальная память представляет расширение адресного пространства задачи, полученное за счет использования части внешней памяти. Windows самостоятельно выбирает размер виртуальной памяти в зависимости от реальной потребности текущей задачи.

Часть виртуального пространства, выделяемого для решения задачи, всегда находится в оперативной памяти. Остальная часть располагается на дисковой памяти. Если оперативной памяти не хватает для работы текущего приложения, то приложение (или его часть), которое не использует в данный момент микропроцессор, выгружается (вытесняется) из оперативной памяти на диск. На освободившееся место загружается (подкачивается) необходимый фрагмент активного приложения. Таким образом, программы циркулируют между диском и оперативной памятью. Поддержка виртуальной памяти позволяет открыть большое количество приложений одновременно, но выгрузка на диск и загрузка с диска снижают производительность компьютера.

# 3.2 Графический пользовательский интерфейс

Графический интерфейс Windows разработан на основе принципа WYSIWYG (What You See Is What You Get — *что видишь, то и получишь*). Это означает, что изображение документа на экране при его просмотре и редактировании и изображение документа на бумаге при его распечатке практически не отличаются.

Пользовательский интерфейс обеспечивает взаимодействие пользователя с персональным компьютером. Символьный интерфейс используется обычно при работе видеосистемы в текстовом режиме, когда информация выводится на экран монитора посимвольно. Графический интерфейс предполагает, что видеосистема может работать в графическом режиме, т. е. выводить на экран монитора информацию поточечно. Графический интерфейс по сравнению с символьным воспринимается как более понятный и интуитивно ясный.

При разработке Windows использовались графические средства: рисунки, специальные значки, цветовое оформление, разнообразные начертания шрифтов, дизайн экрана и др. «Дружественный» по отношению к человеку интерфейс не требует специальных программистских знаний. Принятая концепция графического пользовательского интерфейса характерна для всех программных продуктов под Windows, что обеспечивает комфортную среду работы пользователя.

Работа в графической ОС основана на взаимодействии *активных* (указатель мыши) и *пассивных* графических элементов управления (кнопки, флажки, переключатели, списки и т.п.).

#### 3.3 Оконный интерфейс Windows

Основу графического интерфейса Windows пользователя составляет хорошо организованная система окон и других графических объектов. Общая концепция Windows состоит в максимальной стандартизации всех элементарных приемов работы. Структура окон унифицирована и пользовательский интерфейс однообразен.

# 3.3.1 Стандартные элементы окна приложения

Заголовок окна приложения отображает название приложения. Управляющее (основное) меню содержит имена ниспадающих меню. Ниспадающее меню содержит группы команд, объединенных по функциональному назначению.

*Кнопки Свернуть, Развернуть (Восстановить) и Закрыть;* дублируют команды системного меню и служат для ускорения их вызова.

Заголовок окна документа отображает название документа, совпадающее с именем файла.

Строка состояния содержит информацию о режимах работы приложения. Окно документа всегда встроено в окно приложения. По своей структуре оно напоминает окно приложения, но существенно проще.

*Рабочее поле* предназначено для расположения создаваемых в приложении документов.

Если в данном приложении открыто несколько окон документов, то пользователь может работать в одном из этих окон. Это окно называется активным, его заголовок выделяется цветом. Остальные окна — naccushise.

# 3.3.2 Элементы управления окна диалога

Окно диалога служит для настройки параметров операционной системы или приложения; выводит необходимые в процессе работы сообщения.

Bкладки имеют вид окна диалога; расположены одна под другой, так что видны только их ярлычки.

*Командные кнопки* служат для выполнения написанных на них команд (ОК, ОТМЕНА, ...).

 $\Pi$ ереключатели предназначены для выбора одного из нескольких возможных вариантов; выбранный вариант отмечается точкой внутри круга.

 $\Phi$ лажки предназначены для включения/выключения режима; включенный режим отмечается галочкой внутри квадрата.

*Поле списка* служит для выбора одного варианта из предлагаемого перечня.

Поле ввода служит для ввода текста или числовых данных. Счетчик служит для ввода числовых значений.

# 3.4 Объекты пользовательского уровня

Программные продукты, предназначенные для работы пользователя: *пакет прикладных программ, комплекс прикладных программ, прикладные программы*. Разработчики Windows ввели еще один синоним — *приложение*. Термин «задача» тесно связан с термином «приложение». Приложение становится задачей после его запуска.

Объекты, создаваемые в среде приложения: тексты, таблицы, рисунки, звуки. В среде Windows такие объекты называются документами. Тип файла может задаваться самим приложением по умолчанию. Большинство приложений предлагают пользователю несколько вариантов типа документа. Документы одного типа имеют один и тот же значок, с помощью которого можно легко отличать одни документы от других и узнать приложение, обрабатывающее данный документ по умолчанию.

# 3.5 ОБМЕН ДАННЫМИ В СРЕДЕ WINDOWS

Обмен данными — это передача информации от одного объекта к другому; процесс ввода/вывода данных между оперативной памятью и периферийными устройствами.

**Источник** (сервер) — приложение, откуда производится вставка фрагмента в составной документ.

**Приемник** (клиент, адресат) — приложение, где находится главная часть составного документа, и куда вставляются объекты из других приложений (источников).

**Буфер обмена** — часть виртуальной памяти, которая служит неким перевалочным пунктом при обмене данными; обслуживается операционной системой.

При создании документа иногда возникает необходимость вставить в него фрагменты документов, созданных в других программных средах. Операционная система Windows поддерживает механизмы обмена, с помощью которых любое приложение может включать в свои документы любые данные. Эту универсальность обеспечивает технология OLE.

Документ, содержащий разнотипные данные, получил название составного или интегрированного документа. В составном документе можно выделить главную часть, которая создавалась в одном приложении, и куда вставлялись объекты из других приложений. Составной документ вызывается из приложения, где создавалась его главная часть.

При небольших объемах передаваемых данных для буфера обмена выделяется часть оперативной памяти.

Свойства буфера обмена:

- буфер обмена доступен из любого приложения;
- буфер обмена очищается при перезагрузке операционной системы;
- объект в буфере обмена хранится в формате, определяемом источником;
- при вставке объекта в составной документ, предоставляется возможность его преобразования в другой формат.

Объект вставляется в составной документ как OLE-объект, сохраняющий связь с приложением-источником. Его можно редактировать непосредственно в составном документе, вызвав приложение-источник.

## \*4 ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

#### 4.1 ОСНОВНЫЕ ПОНЯТИЯ

Процедурное программирование является отражением фон Неймановской архитектуры компьютера. Это способ определения и изменения содержимого памяти с помощью команды присвоения.

Функциональное программирование — это способ составления программ, в которых единственным действием является вызов функции.

**Логическое программирование** - это программирование в терминах логики.

**Структурное программирование** позволяет отобразить структуру решаемой задачи на структуру программы и получить ясный алгоритм ее решения.

 ${\tt Команда}$  — это указание где расположены данные, над которыми должно быть выполнено действие на аппаратном уровне.

Операнды — данные, над которыми выполняется операция.

Программа - последовательность команд.

Алгоритм записывается с помощью команд и управляющих конструкций языка программирования. Данные хранятся в ячейках памяти компьютера, к которым обращаются с помощью идентификатора (имени) переменной. Переменная имеет тип — формат хранимой в переменной информации, например, целый, вещественный для чисел или строковый для текста. Присваиваемые переменным значения должны соответствовать объявленному типу. Данные, которые не изменяются в программе, называются константами.

Базовые типы данных — числа, строки, логические величины.

Для работы с целыми числами используются стандартные функции div — деление нацело и mod — остаток от деления нацело. Операции \*, /, div, mod имеют одинаковый приоритет.

Результатом выполнения операции x div y будет целое число, округленное по направлению  $\kappa$  0, например, 17 div 8 = 2, 3 div 5 = 0.

Результатом выполнения операции  $x \mod y = x - (x \operatorname{div} y)^*y$  будет целое число — остаток от деления нацело  $x \bowtie y$ , например, 17  $\mod 8 = 1$ , 3  $\mod 5 = 3$ .

В записи очень больших или очень маленьких чисел используют степень числа 10, например, 5e+20 ( $5*10^{20}$ ) или 3.2e-15 ( $3.2*10^{-15}$ )

Арифметические выражения состоят из арифметических операций и операндов. В качестве операндов используются константы и переменные. В логических выражениях используются операции сравнения и логические операции. Приоритеты логических операций ниже,

чем у операций сравнения.

Строковые данные заключаются в кавычки, которые могут быть парные или одинарные в зависимости от языка программирования. Строковые данные можно сравнивать на совпадение (равно, не равно) и сцеплять друг с другом.

Операторы отделяются друг от друга разделителем, например, точкой с запятой «;». Это нужно для того, чтобы при компиляции различать, где заканчивается один оператор и начинается другой.

Комментарии используются для пояснений к коду программы или для пропуска при компиляции части кода. Синтаксис комментария в одном языке программирования может варьироваться. Например, для однострочного комментария используют сдвоенную наклонную черту, (//комментарий), а для многострочного — фигурные ({комментарий}).

Управляющие структуры позволяют организовать ветвление программы (условный оператор) или выполнение действий в цикле (оператор цикла).

Общая конструкция условного оператора (в квадратных скобках указывается необязательный компонент): *если* условие истинно, *то* выполнить оператор или блок операторов [*иначе* выполнить оператор 2 или блок операторов].

Общая конструкция оператора счетного цикла:  $\partial$ ля начальное значение счетчика  $\partial$ о [шаг приращения счетчика] конечное значение счетчика  $\partial$ елать оператор или блок операторов.

Общая конструкция оператора условного цикла: *пока* условие истинно *делать* оператор или блок операторов.

# 4.2 КОМПИЛЯЦИЯ, ОТЛАДКА И ВЫПОЛНЕНИЕ ПРОГРАММ

**Машинный язык** — внутренний язык ЭВМ, на котором представляется и исполняется программа.

**Исходный модуль** — текст алгоритма задачи, описанный средствами языка программирования.

**Компиляция** (интерпретация) — процедура перевода исходного модуля в последовательность команд ЭВМ.

**Исполняемый файл** — это файл, который может быть выполнен компьютером без предварительной трансляции.

Процедура программирования заключается в реализации алгоритма решения задачи с использованием ЭВМ. Этапы решения задач на ЭВМ приведены на рисунке 1.

# 

Рисунок 1 – Этапы решения задач на ЭВМ

Для получения рабочей программы ее исходный текст необходимо откомпилировать. В процессе компиляции последовательно выполняются лексический, синтаксический, семантический анализ.

При *лексическом анализе* выявляются отельные составляющие текста программы (лексемы) и определяется их тип. К числу лексем относятся названия операторов, имена переменных или констант, разделители (круглые скобки, знаки препинания и т.д.). Каждая лексема заменяется кодом ее типа. Кроме того, из текста программы удаляются пробелы.

Синтаксический анализ служит для определения синтаксической правильности закодированной цепочки лексем. Определяются парность скобок, наличие в нужном месте требуемого разделителя, соответствие исходного текста структурным правилам языка программирования. Например, в операторе условного цикла нельзя опустить условие проверки.

На этапе *семантического анализа* выявляются ошибки, допущенные программистом в нарушение правил составления программ на данном языке программирования.

Тестирование служит для проверки логической правильности программы и гарантирует, что программа функционирует на всем диапазоне допустимых данных. Для тестирования составляются наборы тестов, на которых проверятся работа программы. Тест, приводящий к неправильному выполнению программы, свидетельствует о наличии в программе ошибки. Процесс локализации местонахождения ошибки и ее исправления называется *отладкой*.

*Компилятор* транслирует весь текст исходного модуля в машинный (объектный) код за один непрерывный процесс.

*Компоновщик* собирает программу из объектных модулей и стандартных библиотечных модулей.

*Исполняемый файл* получается в результате компиляции и компоновки объектных модулей и содержит машинные команды и/или команды операционной системы.

Перед выполнением программы ее *объектный модуль* обрабатывается *редактором связей*, создающим *загрузочный модуль*. Загрузчик определяет для загрузочного модуля абсолютные адреса в оперативной памяти, после чего программа может выполняться.

*Интерпретатор* выполняет исходный модуль программы в режиме «оператор за оператором», превращая каждый оператор языка программирования в машинные команды.

Различия: после получения объектного модуля присутствие в оперативной памяти *компилятора* необязательно; в период выполнения исходной программы присутствие в оперативной памяти *интерпретатора* обязательно.

#### 4.3 Объектно-ориентированное программирование

Объект — это абстракция множества предметов реального мира, обладающих общими характеристиками и законами поведения.

**Экземпляр объекта** — это конкретный элемент данного множества.

**Класс** — это множество предметов реального мира, связанных общностью структуры и поведением.

**Экземпляр класса** — это конкретный элемент данного множества.

**Атрибут** — это специальные признаки, посредством которых можно задать правила описания свойств объектов.

Свойство — характеристика объекта, его параметр.

**Метод** — программа действий над объектом или его свойствами.

Событие - изменение состояния объекта.

Полиморфизм — способность объекта принадлежать более чем одному типу.

 ${\tt Hacnegobahue}$  — возможность определения новых классов на основе существующих с возможностью добавления и переопределения данных и методов.

**Инкапсуляция** — процесс отделения друг от друга отдельных элементов объекта, определяющих его устройство и поведение; данные и процедуры объекта скрываются от внешнего пользователя.

Современная технология разработки программных продуктов базируется на концепции *объектно-ориентированного программирования*, в которой выдерживается единый подход к данным и программам. В основе лежит понятие объекта, который объединяет в себе алгоритмы и данные, обрабатываемые этими алгоритмами.

Характеристики объектно-ориентированного программирования (Алан Кэй):

- все в компьютерном мире является объектами;
- вычисление в компьютере это обмен данными между объектами. Объекты взаимодействуют, посылая и получая сообщения запрос на выполнение действия;
  - каждый объект имеет независимую память;
  - память объекта состоит из других объектов;
  - каждый объект является представителем класса;
- каждый класс выражает свойства принадлежащих ему объектов;
- в классе задается поведение объекта, поэтому все объекты, принадлежащие к данному классу, могут выполнять одинаковые действия;
- все классы образуют структуру, отражающую *иерар-хию наследования*. Память и поведение, связанное с экземплярами определенного класса, могут использоваться любым классом, расположенным ниже в иерархической структуре.

Класс, от которого произошло наследование, называется *базовым* или *родительским*. Классы, которые произошли от базового, называются *потомками*, *наследниками* или *производными классами*.

Операционная система Windows создана на базе объектноориентированной методологии программирования. Основные понятия рабочей среды: *объект, его свойства и действия*, которые объект может выполнять в зависимости от запроса.

В объектно-ориентированной среде с любым объектом сопоставлена определенная совокупность действий. Последовательность действий в среде Windows приведена на рисунке 2.



Рисунок 2 – Последовательность действий в среде Windows

Объектно-ориентированная технология Windows предоставляет возможность пользователю создавать документы, фрагменты которых подготовлены в разных средах.

#### 4.4 ОБЗОР ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Языки программирования отличаются от естественных строгими правилами записи команд — синтаксисом и семантикой — смыслом используемых в языке конструкций. С помощью языка программирования создается текст программы — исходный модуль.

Деление языков программирования на классы представлено на рисунке 3.

Языки программирования низкого уровня — ассемблеры — ориентированы на конкретный тип процессора и позволяют писать программы в машинных кодах. На них пишут небольшие системные программы, драйверы устройств.

Языки программирования высокого уровня ближе к естественным языкам и не учитывают особенности компьютерной архитектуры.

Основная идея *процедурного программирования* — использование памяти для хранения данных. Основная команда — *присвоение*, с помощью которой определяется и меняется память компьютера. Программа производит преобразование содержимого памяти, изменяя его от исходного состояния к результирующему.

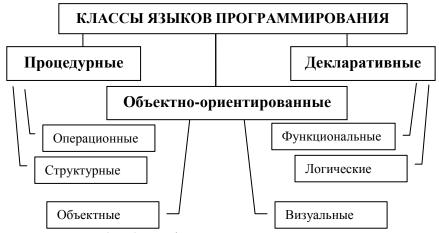


Рисунок 3 – Классификация языков программирования

# 4.4.1 Языки процедурного программирования

Фортран (начало 1950-х годов) — первый компилируемый язык для программирования научно-технических задач.

Алгол (1960 г.) — это многоцелевой расширенный язык программирования. В нем впервые введены понятия «блочная структура программы» и «динамическое распределение памяти». Из-за сложной структуры не получил широкого распространения.

Кобол (начало 1960-х годов) создан для решения бизнес-задач, связанных с обработкой больших объемов данных, хранящихся на различных носителях.

Бейсик (середина 1960-х годов) — специализированный учебный язык программирования для начинающих. Характеризуется простотой освоения и наличием универсальных средств для решения научных, технических и экономических задач. По популярности занимает одно из первых мест в мире.

PL-1 (1963 – 1966 гг.) — многоцелевой универсальный язык, хорошо приспособленный для исследования и планирования вычислительных процессов, моделирования, решения логических задач, разработки систем математического обеспечения. Позволяет задавать точность вычислений. Поддерживается компанией IBM.

 $\Pi$ аскаль (1968 — 1971 гг.). Основные принципы языка  $\Pi$ аскаль приведены на рисунке 4.

#### ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ

# Структурный (использование подпрограмм и независимых структур данных) пр

# «Сверху-вниз»

(решение исходной задачи путем ее деления на мелкие и простые подзадачи)

Рисунок 4 – Принципы программирования в языке Паскаль

C (Си, начало 1970-х годов) планировался как язык для реализации операционной системы Unix вместо языка Ассемблер. Одна из особенностей языка Си — сглаживание различий между выражениями и операторами, что приближает его к функциональным языкам программирования. В языке отсутствует понятие процедуры. Использование подпрограмм основано на понятии функции, которая может сочетать в себе возможности процедуры. По набору управляющих конструкций и структур данных язык Си можно отнести к языкам высокого уровня. Однако набор средств прямого обращения к функциональным узлам компьютера позволяет использовать Си как операционный язык. Язык Си во многом похож на Паскаль.

# 4.4.2 Языки объектно-ориентированного программирования

Наиболее современными языками программирования являются C++ и Java. C++ (1980 г.) является расширением языка Си.

Java (Ява, начало 1990-х годов) создан для упрощения разработки приложений на C++ за счет исключения низкоуровневых возможностей. Программа компилируется не в машинный код, а в программно-независимый байт-код. Каждая команда программы занимает 1 байт и выполняется с помощью интерпретатора JVM — виртуальной Java-машины. Для разных аппаратных платформ созданы свои Java-машины, что позволяет переносить программы на уровне двоичного байт-кода.

C# (Си Шарп, конец 1990-х годов) объединил в себе лучшие идеи языков С, С++, Java.

С середины 90-х годов в объектно-ориентированных языках реализуются *системы визуального проектирования*. Эти системы позволяют создавать интерфейсную часть программного продукта в диалоговом режиме. К объектно-ориентированным системам визуаль-

ного проектирования относятся Visual Basic, Delphi, C++ Builder, Visual C++.

VBA — язык приложений Microsoft Office (Excel, Word, Power Point и др). VBA соблюдает основной синтаксис языка и правила программирования языков Basic-диалектов. Позволяет создавать макросы для автоматизации выполнения некоторых действий.

# 4.4.3 Языки декларативного программирования

В функциональном программировании не используется память для хранения данных, промежуточные переменные, операторы присваивания и циклы. Ключевым понятием в функциональных языках является выражение. Программа, написанная на функциональном языке, представляет собой последовательность описания функций и выражений. Выражение вычисляется сведением сложного к простому. Все выражения записываются в виде списков.

 $\it Лисп~(1959~г.)$  позволяет обрабатывать большие объемы текстовой информации.

Пролог (1973 г.) — язык искусственного интеллекта. Программа строится из последовательности фактов и правил. Затем формулируется утверждение, которое Пролог пытается доказать с помощью правил. Язык сам ищет решение с помощью методов поиска и сопоставления, которые в нем заложены. Логические программы не отличаются высоким быстродействием, так как процесс их выполнения сводится к построению прямых и обратных цепочек рассуждений.

#### 4.4.4 Языки баз данных

Языки баз данных отличаются от алгоритмических языков своим функциональным назначением. Для обработки больших массивов информации и выборки записей по определенным признакам был создан структурированный язык запросов SQL (Structured Query Language). SQL является стандартным языком для работы с *реляционными* базами данных, позволяет манипулировать не отдельными записями, а группами записей.

СУБД — системы управления базами данных— поддерживают язык SQL. Ведущие производители СУБД: Microsoft (SQL Server), IBM (DB2), Oracle, Software AG, Informix и Sybase. В каждой СУБД имеется свой встроенный язык работы с базами данных. Например, в Oracle имеется встроенный язык PL/SQL, в Informix — INFORMIX 4GL, в Adabas — Natural.

# 4.4.5 Языки программирования для компьютерных сетей

Языки программирования для компьютерных сетей являются *интерпретируемыми*. Интерпретаторы для них распространяются бесплатно, а сами программы — в исходных текстах. Такие языки называются *скрипт-языками*.

*HTML* (Hyper Text Markup Language) — универсальный язык разметки гипертекста, используемый для подготовки веб-документов. Язык позволяет форматировать текст, добавлять графические образы, работать с таблицами и организовывать связь одного документа с другим с помощью механизма гиперссылок.

Perl (1980 г.) содержит многочисленные функции для работы со строками, массивами, всевозможные средства преобразования данных. Позволяет обрабатывать большие текстовые файлы, генерировать текстовые отчеты.

PHP~(1995-1997~гг.) — это улучшенный Perl со встроеными средствами доступа к базам данных. Используется для создания динамических сайтов.

VRML (1994 г.) создан для организации виртуальных трехмерных интерфейсов в Интернете.

XML (1996 г.) — универсальный язык разметки структуры документов. Альтернатива языка HTML.

# Вопросы для самоконтроля

- 1. Дать понятие алгоритма.
- 2. Перечислить основные свойства алгоритма.
- 3. Из каких базовых элементов составляется алгоритм?
- 4. Зачем нужны трансляторы?
- 5. Чем компилятор отличается от интерпретатора?
- 6. Дать понятие файловой системы.
- 7. Перечислить этапы решения задач на ЭВМ.
- 8. Какие основные понятия используются в объектно-ориентированном программировании?
- 9. Перечислить классы языков программирования.
- 10. Какие принципы программирования используются в процедурных языках?

# Тема 5 ПРЕДСТАВЛЕНИЕ, КОДИРОВАНИЕ, КОНТРОЛЬ ПЕРЕДАЧИ ИНФОРМАЦИИ

# 1 СПОСОБЫ ПРЕДСТАВЛЕНИЯ И МЕТОДЫ ПЕРЕДАЧИ ИНФОРМАЦИИ ПО КАНАЛАМ СВЯЗИ

# 1.1 ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ ФИЗИЧЕСКИМИ СИГНАЛАМИ

**Такт** - временной интервал между двумя соседними моментами дискретного времени.

Так как информация представляется в двоичном алфавите, то физическими аналогами знаков 0 и 1 служат сигналы, способные принимать два хорошо различимых значения. Например, напряжение высокого и низкого уровней, отсутствие и наличие электрического импульса, и т.п. В схемах цифровых устройств сигналы изменяются в дискретные моменты времени (0, 1, 2,...,). Специальный блок вырабатывает синхронизирующие сигналы, которые отмечают моменты дискретного времени (границы тактов).

# Способы физического представления информации

- 1. Потенциальный. Двум значениям переменной 1 и 0 соответствуют разные уровни напряжения потенциальный код. Потенциальный сигнал сохраняет постоянный уровень в течение такта; его значение в переходные моменты не является определенным.
- 2. *Импульсный*. Двум значениям двоичной переменной 1 и 0 соответствует наличие и отсутствие электрического импульса либо разнополярные импульсы импульсный код.

По типам используемых сигналов для представления информации схемы цифровых устройств делятся на *импульсные*, *потенциальные* и *импульсно-потенциальные*.

# 1.2 МЕТОДЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

При передаче информации используются коды:

1. Последовательный. Каждый временной такт предназначен для отображения одного разряда кода слова. Все разряды слова фиксируются по очереди одним и тем же элементом и проходят через одну линию передачи информации.

2. Параллельный. Все разряды кода слова представляются в одном временном такте, фиксируются отдельными элементами и проходят через отдельные линии. Каждая линия служит для передачи только одного разряда слова. Таким образом, код слова развертывается не во времени, а в пространстве, так как значения всех разрядов слова передаются по нескольким линиям одновременно.

# 1.2.1 Синхронная передача

При *синхронной* передаче блок символов передается непрерывно в принудительном темпе. Синхронизация передающего и принимающего устройств достигается посылкой специальных кодовых комбинаций перед каждым блоком данных.

Передающее устройство устанавливает одно из двух возможных состояний сигнала (0 или 1) и поддерживает его в течение определенного времени, по истечении которого состояние сигнала на передающей стороне может быть изменено. Время передачи сигнала складывается из времени распространения сигнала по линии и времени распознавания и фиксации сигнала в регистре приемного устройства. Пусть T — максимальное время передачи сигнала с учетом наихудших условий. Тогда для синхронной передачи информации должно выполняться условие  $\tau \geq T$ , где  $\tau$  — период, в течение которого передатчик поддерживает состояние сигнала.

\*Передача со стробированием. Информация передается по линиям  $\Pi_1, ..., \Pi_n$  в интервале времени, когда сигнал на линии  $\Pi_0$  соответствует 1. При нулевом сигнале на линии  $\Pi_0$  сигналы на шинах  $\Pi_1, ..., \Pi_n$  не имеют смысла. Для гарантированной передачи данных по линиям  $\Pi_1, ..., \Pi_n$  передаваемый код устанавливается как минимум на время  $\Delta T$  раньше, чем появляется единичный сигнал на линии  $\Pi_0$ . Снятие передаваемого кода с линий  $\Pi_1, ..., \Pi_n$  может закончиться на время  $\Delta T$  позже времени установления нулевого состояния на линии  $\Pi_0$ . Передача со стробированием используется главным образом для пересылок информации внутри устройства, например, между регистрами.

# 1.2.2 Асинхронная передача

При *асинхронной* передаче символы передаются в свободном темпе независимо друг от друга. Каждый символ передается со своими сигналами "Старт" и "Стоп", указывающими на начало и конец передачи символа. Асинхронная передача позволяет передавать информацию с устройств, которые выдают ее асинхронно во времени (на-

пример, клавиатура).

Передающее устройство устанавливает соответствующее передаваемому коду состояние сигнала на линии. Принимающее устройство после приема сигнала информирует об этом передатчик, изменяя состояние сигнала на линии. Передающее устройство, получив сигнал о приеме, снимает передаваемый сигнал. Таким образом, период  $\tau$ , является переменным и зависит от характеристик конкретной линии связи и устройств, участвующих в передаче.

Пусть t — время передачи нового состояния сигнала в один конец линии связи. Тогда при асинхронной передаче должно выполняться условие  $\tau \geq 2t$ . Обычно время 2t значительно меньше времени T, которое выбирается, исходя из максимально возможных расстояний между устройствами. При передаче параллельного кода по параллельным линиям сигналы поступают в приемное устройство в разное время из-за разброса параметров цепей, формирующих сигналы, и линий интерфейса. Максимальный разброс времени передачи  $\Delta T = \max\{|t_i - t_i|\}$ 

\*Передача с квитированием. Сигнал на линии  $\Pi_0$  сигнализирует приемнику, что передатчик подготовил передаваемую информацию на линиях  $\Pi_1$  ...,  $\Pi_n$ . Сигнал на линии  $\Pi_{n+1}$  сигнализирует передатчику, что приемник принял передаваемую информацию. Приняв сигнал по линии  $\Pi_{n+1}$ , передатчик снимает информацию с линий  $\Pi_1$ , ...,  $\Pi_n$  и гасит сигнал на линии  $\Pi_0$ . Отсутствие сигнала на линии  $\Pi_0$  сигнализирует приемнику об окончании передачи данных. После этого приемник гасит сигнал на линии  $\Pi_{n+1}$ . Отсутствие сигнала на линии  $\Pi_{n+1}$  сигнализирует передатчику о готовности приемника к приему следующей порции данных. Передача с квитированием используется, когда приемное устройство не всегда готово к приему информации (занято выполнением других операций).

#### \*1.3 ХАРАКТЕРИСТИКА КАНАЛОВ СВЯЗИ

**Канал связи** — совокупность средств, обеспечивающих передачу сигналов; физическая среда и аппаратурные средства, осуществляющие передачу информации от одного узла коммутации к другому либо к абоненту связи.

Физическая среда — пространство или материал, обеспечивающие распространение сигналов (проводная воздушная или кабельная линия, скрученная пара проводов, коаксиальный кабель, стекловолоконная линия, эфир).

Канал информационный — это совокупность устройств, объединенных линиями связи, предназначенных для передачи информации от источника информации (начального устройства канала) до ее приемника (конечного устройства канала).

**Задержка сигнала во времени** — это интервал времени от отправки сигнала передатчиком до его приема приемником.

**Канал передачи данных** — канал связи, оснащенный аппаратурой для передачи дискретной информации.

Примеры каналов связи и предельных частот их работы телеграф —  $140\Gamma$ ц; телефон — до 3.1К $\Gamma$ ц; короткие волны (10-100м) — 3-30М $\Gamma$ ц; УКВ (1-10м) — 30-300М $\Gamma$ ц; спутник (сантиметровые волны) — до  $30\Gamma$  $\Gamma$ ц; оптический (инфракрасный диапазон) — 0.15-400Т $\Gamma$ ц; оптический (видимый свет) — 400-700Т $\Gamma$ Ц; оптический (ультрафиолетовый диапазон) — 0.7-1.75П $\Gamma$ ц.

По телеграфным каналам информация передается в дискретной форме, что облегчает их сопряжение с ЭВМ. По телефонным каналам информация передается в аналоговой форме, что усложняет сопряжение этих каналов с ЭВМ.

Технические характеристики канала определяются принципом действия входящих в него устройств, видом сигнала, свойствами и составом физической среды, в которой распространяются сигналы, свойствами применяемого кода.

Эффективность канала характеризуется скоростью и достоверностью передачи информации, надежностью работы устройств и задержкой сигнала во времени.

Математически канал задается множеством допустимых сообщений на входе, множеством допустимых сообщений на выходе и набором условных вероятностей P(y/x) получения сигнала y на выходе при входном сигнале x. Условные вероятности описывают статистические свойства "шумов" (или помех), искажающих сигнал в процессе передачи. В случае, когда P(y/x) = 1 при y = x и P(y/x) = 0 при y <> x, канал называется  $\kappa$  каналом без "шумов".

В соответствии со структурой входных и выходных сигналов выделяют дискретные и непрерывные каналы. В *дискретных каналах* сигналы на входе и выходе представляют собой последовательность символов одного или двух (по одному для входа и выхода) алфавитов.

В непрерывных каналах входной и выходной сигналы представляют собой функции от непрерывного параметра-времени.

Каналы различаются по направлениям передачи информации (рисунок 1).



Рисунок 1 – Типы каналов по направлению передачи информации

# \*1.4 ПЕРЕДАЧА ИНФОРМАЦИИ ПО КАНАЛАМ СВЯЗИ

**Шум** — это помехи в канале связи при передаче информации.

**Кодирование** — преобразование дискретной информации одним из следующих способов: шифрование, сжатие, защита от шума.

**Линия связи** — это среда, по которой передаются сигналы от передатчика к приемнику.

Различные по физической природе сообщения (цифровые данные, речь, результаты измерений различных физических величин) предварительно должны быть преобразованы в электрические колебания, сохраняющие все свойства исходных сообщений. Для экономного использования линии связи и уменьшения влияния помех передаваемая информация может быть преобразована с помощью кодирующего устройства (рисунок 2).

Преобразование включает статистическое помехоустойчивое кодирование. В результате на выходе кодирующего устройства образуется последовательность, которую передатчик преобразует в форму, удобную для передачи по линии связи.



Рисунок 2 – Передача информации по каналу связи

На вход приемника попадают сигналы и различные помехи. Приемник выделяет из смеси сигнала и помех последовательность, которая должна соответствовать последовательность на выходе кодирующего устройства. Но из-за действия помех и влияния среды полное соответствие получить невозможно. Полученная последовательность вводится в декодирующее устройство, которое преобразует ее в последовательность, соответствующую переданной. Полнота этого соответствия зависит от корректирующих возможностей кодированной последовательности, уровня сигнала и помех, свойств кодирующего устройства.

Пусть передатчик описывается случайной величиной x, тогда из-за помех в канале связи на приемник будет приходить случайная величина y = x + z, где z — это случайная величина, описывающая помехи. Можно говорить о количестве информации, содержащейся в случайной величине y, относительно x. Чем ниже уровень помех, тем больше информации можно получить из y. При отсутствии помех y содержит в себе всю информацию об x.

Основным показателем качества передачи является достоверность информации. Достоверность передачи информации характеризуется коэффициентом ошибок:

$$K_{\text{ош.}} = N_{\text{ош.}}/N$$
,

где  $N_{\text{ош.}}$  — количество неверно принятых сообщений; N — общее количество переданных сообщений.

При увеличении N коэффициент ошибок стремится к вероятности ошибки:

$$P_{ou.} = \lim_{\substack{N \to \infty \\ N \to \infty}} / N$$

При передаче по каналу связи информация подвергается воздействию помех (рисунок 3).

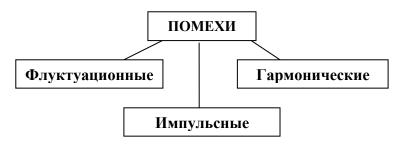


Рисунок 3 – Типы помех в каналах связи

Флуктуационная помеха представляет собой напряжение, меняющееся во времени случайным образом; причина — тепловые шумы линии, элементов аппаратуры: действуют непрерывно в течение длительного времени.

Гармоническая помеха приближенно описывается синусоидальным колебанием; возникает в самой аппаратуре из-за проникновения в канал различных несущих колебаний; действует непрерывно

*Импульсной помехой* (рисунок 4) называется помеха, пиковое значение которой соизмеримо с амплитудой полезного сигнала или превышает ее; появляются пачками (по несколько помех в пачке). Действуют на сигнал в отдельные моменты времени.



Рисунок 4 — Разновидности импульсных помех по типу источника

#### 1.5.1 Передача информации по каналу без помех

Пусть через канал связи без помех передается последовательность дискретных сообщений длительностью Т. Тогда скорость передачи информации по каналу связи будет:

$$\lim(I/T) = V[\acute{a}\grave{e}\grave{o} / \tilde{n}],$$

$$T \rightarrow \infty$$

где I — количество информации, содержащейся в последовательности сообщений. Предельное значение скорости передачи информации — пропускная способность канала связи — определяется выражением:

$$C = V_{\text{max}} = \lim_{T \to \infty} \left( I_{\text{max}} / T \right).$$

Скорость передачи информации измеряется в количестве переданных за одну секунду бит или в бодах: 1бод = 1бит/сек (bps). Пропускная способность не зависит от скорости передачи информации и выражается максимальным количеством двоичных единиц информации, которое данный канал связи может передать за одну секунду. Основное условие согласования источника информации и канала связи:  $V \le C$ . Согласование осуществляется путем кодирования сообшений.

Способность канала передавать информацию характеризуется числом — *пропускной способностью* или *емкостью канала* (C).

Емкость канала без помех:

$$C = \lim_{T \to \infty} \frac{\log_2 N(T)}{T},$$

где N(T) — число всех возможных сигналов за время T.

**Пример**. Пусть алфавит канала без "шумов" состоит из двух символов — 0 и 1, длительность  $\tau$  секунд каждый. За время T успеет пройти  $n=T/\tau$  сигналов, всего возможны  $2^{\rm n}$  различных сообщений длиной n.

$$C=\lim_{T o\infty}rac{\log_22^{T/ au}}{T}=1/ au$$
 бод

На рисунке 5 приведена схема процесса прохождения информации по каналу с описанными в примере характеристиками.

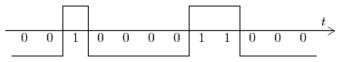


Рисунок 5 – Прохождение информации по каналу без помех

Здесь для кодирования 0 используется низкий уровень сигнала, для 1 — высокий.

#### 1.5.2 Передача информации по каналу с помехами

При передаче информации через канал с помехами сообщения искажаются; количество получаемой информации уменьшается на величину неопределенности, вносимой помехами:

$$I = H(i) - H_i(i),$$

где H(i) — энтропия источника сообщений;  $H_j(i)$  — энтропия сообщений на приемной стороне.

Выражение для скорости передачи по каналу связи с помехами:

$$V' = \lim_{T \to \infty} (I/T) = \lim_{T \to \infty} ((H_i - H_j(i))/T).$$

Если энтропия источника информации не превышает пропускной способности канала ( $H \ll C$ ), то существует код, обеспечивающий передачу информации через канал с помехами со сколь угодно малой частотой ошибок.

Пусть  $W_c$  — средняя мощность сигнала;  $W_{\rm m}$  — средняя мощность помех. Тогда для канала с высоким уровнем шумов ( $W_{\rm m}>>W_c$ ) максимальная скорость передачи близка к нулю.

#### \*1.6 КОММУТАЦИЯ В СЕТЯХ

Уплотнение каналов связи — статическое разделение канала, при котором определенные полосы частот (или периоды времени) выделяются в фиксированном порядке для использования в качестве отдельных каналов.

**Концентрация каналов связи** — динамическая процедура распределения меньшего числа более скоростных выходных каналов между большим числом менее скоростных входных каналов.

Коммутация каналов связи — совокупность операций по соединению каналов для получения сквозного канала, который связывает через узлы коммутации один абонентский пункт с другим.

Коммутация сообщений - передача информации с запомина-

нием в промежуточных узлах сети передачи данных без установления физического соединения между пунктами отправления и назначения.

Виртуальное соединение — это коммутация каналов не физическая, а через память компьютеров в центрах коммутации.

**Датаграмма** — самостоятельный пакет, движущийся по сети независимо от других пакетов.

**Трафик** — нагрузка, создаваемая потоком вызовов, требований, сообщений, поступающих на входы системы массового обслуживания.

Распределением информационных потоков занимается система коммутации (рисунок 6).



Рисунок 6 – Состав системы коммутации

Узлы коммутации обеспечивают установление, поддержание и разъединение соединений между терминалами (телефонный аппарат, компьютер).

Метод коммутации пакетов отличается от метода коммутации сообщений тем, что длинные сообщения разбиваются на части — пакеты.

# 1.6.1 Соединение с накоплением информации

Сигналы из входящих каналов записываются в ЗУ, а оттуда через определенное время поступают в исходящие каналы. Если в момент прихода сигнала по входящему каналу требуемый исходящий канал занят, то возможны два решения:

- 1. Источник сообщений уведомляется о невозможности установления соединения в данный момент (это система с отказами); попытки повторяются до получения положительного результата.
- 2. Входящее сообщение запоминается и передается в исходящий канал по его освобождении (это система с ожиданием).

#### 1.6.2 Коммутация каналов

Устанавливается физическое соединение между пунктами отправления и назначения (источником и адресатом) путем образования составного канала из последовательно соединенных отдельных канальных участков. Пункт отправления посылает сигнализирующее сообщение, которое, перемещаясь по сети передачи данных от одного узла коммутации каналов к другому, прокладывает путь от источника к пункту назначения. Этот путь (составной канал) состоит из физических каналов, имеющих одну и ту же скорость передачи данных. Из пункта назначения в источник посылается сигнал обратной связи. Затем из источника передается сообщение по установленному пути с одновременным использованием всех образующих его каналов. В это время каналы недоступны для других передач.

# 2 КОДИРОВАНИЕ И КОНТРОЛЬ ПЕРЕДАЧИ ИНФОРМАЦИИ

#### 2.1 Основные понятия

**Кодирование** — процесс преобразования сообщений в комбинации из дискретных сигналов.

 ${\bf Kog}$  — совокупность правил, в соответствии с которыми производится кодирование.

Характеристики сигналов кода приведены на рисунке 1.

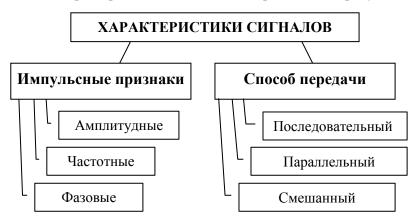


Рисунок 1 – Характеристики сигналов кода

В реальных условиях непрерывный сигнал может быть заменен дискретным. Формой представления информации является сообщение. Каждому сообщению однозначно соответствует определенная кодовая комбинация. Набор элементов данного кода — алфавит, а кодовые комбинации из этих элементов —кодовые слова.

Выбор методов кодирования зависит от количества передаваемых сообщений, требуемого времени передачи, параметров канала связи, возможности аппаратуры.

#### \*2.2 Классификация двоичных кодов

Классификация двоичных кодов приведена на рисунке 2.

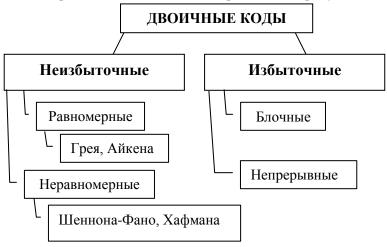


Рисунок 2 – Классификация двоичных кодов

В равномерных кодах все слова содержат одинаковое число разрядов. В неравномерных кодах число разрядов в словах может быть различным. В вычислительных машинах применяются преимущественно равномерные коды.

В *блочных* кодах каждому сообщению соответствует кодовая комбинация (блок) из n символов. Блоки кодируются и декодируются отдельно друг от друга.

Все *избыточные* коды делятся на два класса: непрерывные и блочные (рисунок 3).

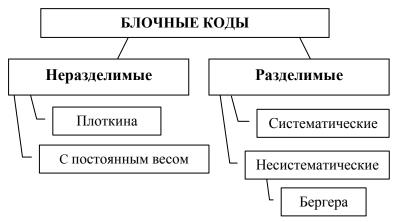


Рисунок 3 – Классификация избыточных блочных кодов

Разделимые коды содержат постоянное число информационных разрядов, представляющих передаваемую информацию, и избыточных. Избыточные разряды занимают одни и те же позиции в кодовом слове. В неразделимых кодах разряды кодового слова невозможно разделить на информационные и избыточные.

Большой класс разделимых блочных кодов составляют *систе- матические коды* (рисунок 4).

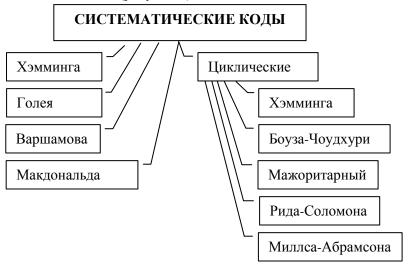


Рисунок 4 – Классификация блочных систематических кодов

Каждый проверочный символ выбирается таким образом, чтобы его сумма по модулю два с определенными информационными символами была равной нулю.

# 2.3 ХАРАКТЕРИСТИКИ КОДОВ

 $\ensuremath{\mathcal{L}}$ лина кода n — число разрядов (символов), составляющих кодовую комбинацию.

Основание кода m — количество отличающихся друг от друга значений импульсных признаков (0 и 1), используемых в кодовых комбинациях. Для случая двоичных кодов m=2.

 $Mощность кода N_p$  — число кодовых комбинаций, используемых для передачи сообщений.

Полное число кодовых комбинаций N — число всех возможных комбинаций длины n из m различных символов, равное  $m^n$  (для двочных кодов  $N=2^n$ ).

Число информационных символов k — количество символов (разрядов) кодовой комбинации, предназначенных для передачи собственно сообщения.

 $\mathit{Число}\ \mathit{npoверочныx}\ \mathit{символов}\ \mathit{r}$  — количество символов (разрядов) кодовой комбинации, необходимых для коррекции ошибок.

Под избыточностью кода R отношение числа проверочных символов к длине кода: R = r/n

*Скорость передачи кодовых комбинаций* — отношение числа информационных символов к длине кода:  $R^{'}=k/n$ .

 $\mathit{Bec}\ \kappa o do so \ i \ \kappa o m \delta u + a u u u \ (\kappa o d a) \ w$  — количество единиц в кодовой комбинации.

Кодовое расстояние *д между двумя кодовыми комбинациями* — число одноименных разрядов с различными символами. Кодовое расстояние выражается как вес суммы кодовых комбинаций.

**Пример 1**. Определить кодовое расстояние между комбинациями 100101 и 001001. Просуммируем комбинации по модулю два:

100101

+

001001

101100

Полученная новая кодовая комбинация имеет вес w=3. Кодовое расстояние между исходными комбинациями d=3.

Весовая характеристика кода W(w) — число кодовых комбинаций веса w.

**Пример 2**. Даны кодовые комбинации 00000, 11110, 11101 и 11010, весовая характеристика W(0) = 1, W(3) = 1, W(4) = 2, т. е. данный код состоит из одного кодового слова веса 0, двух слов веса 4 и одного слова веса 3.

Вероятность необнаруженной ошибки — это вероятность события, при котором свойства данного кода не позволяют определить факт наличия ошибки в принятой комбинации.

Оптимальность кода — свойство кода, которое обеспечивает наименьшую вероятность не обнаружения ошибки среди всех кодов той же длины и избыточности г.

#### \*2.4 МЕТОДЫ КОДИРОВАНИЯ ИНФОРМАЦИИ

**Битовый элемент** — минимальный интервал времени между битами данных, получаемых при постоянной скорости вращения диска

Различают методы кодирования (рисунок 5) и методы записи информации. Методы кодирования взаимно несовместимы, но подразумевают единый метод записи — без возврата к нулю с инверсией. Методы кодирования не влияют на изменения направления тока в магнитной головке при записи, а лишь задают их очередность.



Рисунок 5 – Методы кодирования информации

Кодирование с применением *FM*-сигналов принято называть кодированием с единичной плотностью. В начале битовых элементов записываются биты синхронизации, а в промежутках между ними би-

ты данных. Бит данных записывается через 4 мкс после бита синхронизации. Недостаток: почти половина полезной емкости используется для записи служебной информации. Для жестких магнитных дисков этот метод не применяется.

Метод *MFM* позволяет вдвое увеличить продольную плотность записи. Длительность битового элемента 4 мкс. Биты синхронизации записываются, если в предыдущем и текущем битовых элементах не были записаны биты данных.

При методе  $M^2FM$  изменяется направление тока в случае длинной последовательности нулей на каждые два битовых элемента. Это затрудняет синхронизацию и не дает ощутимого выигрыша в емкости диска и скорости передачи данных. В настоящее время метод не используется.

Метод *RLL* впервые был использован в цифровой записи на магнитную ленту. Каждый байт данных делится на 2 полубайта, которые кодируются 5-разрядным кодом. Эти группы подбираются таким образом, чтобы при передаче данных нули не встречались подряд более двух раз, что делает код самосинхронизирующимся.

**Пример.** Тетрада 0000 заменяется группой бит 11001, тетрада 1000 — 11010, тетрада 0001 — 11011, тетрада 1111 — 01111.

При считывании каждые две 5-разрядные группы декодируются и полубайты объединяются.

*Memod ARLL* — модифицированный RLL. Наряду с уплотнением данных увеличена скорость обмена данными между накопителями и контролером.

Кодирование MFM или FM можно представить как частный случай RLL.

# 2.5 КОНТРОЛЬ ПЕРЕДАЧИ ИНФОРМАЦИИ

Процедуры кодирования и декодирования могут повторяться много раз. Ошибки при передаче информации происходят из-за шума в канале (атмосферные и технические помехи), а также при кодировании и декодировании. Схема передачи информации приведена на рисунке 6. Теория информации изучает, в частности, способы минимизации количества таких ошибок.

КОДИРОВАНИЕ КАНАЛ СВЯЗИ ДЕКОДИРОВАНИЕ ИСТОЧНИК  $\stackrel{\downarrow}{\longrightarrow}$  ПЕРЕДАТЧИК  $\stackrel{\downarrow}{\longrightarrow}$  ПРИЕМНИК  $\stackrel{\downarrow}{\longrightarrow}$  ПОЛУЧАТЕЛЬ.

Достоверность передачи данных оценивается отношением числа ошибочно принятых символов к общему числу переданных. Методы и средства, повышающие достоверность передачи, приведены на рисунках 7— 8.



Рисунок 7 – Методы контроля передачи информации

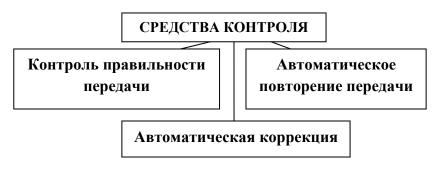


Рисунок 8 – Средства контроля передачи информации

# 2.6 Помехоустойчивое кодирование

Если все разряды слова служат для представления информации, код называется **простым**.

**Минимальное кодовое расстояние** — минимальное расстояние между двумя любыми словами в этом коде.

Для минимизации вероятности ошибки при передаче данных используют *помехозащитные* коды. Идея состоит в добавлении к символам исходных кодов нескольких контрольных символов. При контроле передачи информации наибольшее распространение получили методы информационной избыточности, использующие коды с обнаружением и коррекцией ошибок.

Если длина кода n разрядов, то таким двоичным кодом можно представить максимум  $2^n$  различных слов.

Способность кода обнаруживать или исправлять ошибки определяется так называемым минимальным кодовым расстоянием.

Простой код имеет минимальное расстояние  $d_{min}=1$  (имеется хотя бы одна пара слов, отличающихся друг от друга только в одном разряде).

Для избыточных кодов  $d_{min} > 1$ .

Если  $d_{min} \geq 2$ , то любые два слова в данном коде отличаются не менее чем в двух разрядах. Следовательно, любая одиночная ошибка приведет к появлению запрещенного слова и может быть обнаружена.

Если  $d_{min}=3$ , то любая одиночная ошибка создает запрещенное слово, отличающееся от правильного в одном разряде, а от любого другого разрешенного слова — в двух разрядах. Заменяя запрещенное слово ближайшим к нему разрешенным словом можно исправить одиночную ошибку.

Для того, чтобы избыточный код позволял обнаруживать ошибки кратностью r, должно выполняться условие:  $d_{min} \ge r+1$ . Одновременная ошибка в r разрядах слова создает новое слово. Чтобы оно не совпало с каким-либо другим разрешенным словом, минимальное расстояние между двумя разрешенными словами должно быть хотя бы на единицу больше, чем r.

# 2.6.1. Код с проверкой четности

Контроль четности — простейший код для борьбы с шумом. Код образуется добавлением к группе информационных разрядов, представляющих простой код, одного избыточного разряда. При формировании кода слова в контрольный разряд записывается 0 или 1 таким образом, чтобы сумма 1 в слове, включая избыточный разряд, была четной (при контроле по четности) или нечетной (при контроле по нечетности). В дальнейшем при всех передачах слово передается вместе со своим контрольным разрядом.

Если при передаче информации приемное устройство обнаруживает, что в принятом слове значение контрольного разряда не соответствует четности суммы слова, то это считается признаком ошибки.

Минимальное расстояние кода с проверкой четности  $d_{min}=2$ . Поэтому код обнаруживает все одиночные ошибки и все случаи нечетности числа ошибок (3,5 и т. д.).

Код с проверкой четности применяется для контроля передач информации между регистрами и для контроля считываемой информации в оперативной памяти. При контроле по четности проверяется правильность передачи и отсутствие сбоев при хранении числа в регистре в течение сколь угодно большого времени. При одновременном возникновении четного числа ошибок код не обнаруживает их.

При контроле по нечетности контролируется полное пропадание информации (слово, состоящее из 0, относится к запрещенным).

# 2.6.2. Контроль по совпадению

После передачи информации из одного регистра в другой правильность передачи можно проверить поразрядным сравнением содержимого всех разрядов регистров. При этом не требуется формирования каких-либо дополнительных контрольных разрядов. При контроле передачи по совпадению обнаруживаются ошибки любой кратности, а затраты оборудования меньше, чем при контроле по четности.

Недостаток: позволяет проверять правильность передачи числа в регистр и отсутствие сбоев при его хранении только до тех пор, пока не изменит своего состояния регистр, из которого передавалась информация.

#### 2.6.3. Код Хэмминга

Код Хэмминга строится таким образом, что к имеющимся информационным разрядам слова добавляется определенное число контрольных разрядов, после чего вся конструкция записывается в  $O\Pi$ .

При считывании слова контрольная аппаратура образует из прочитанных информационных и контрольных разрядов корректирующее число, которое равно 0 при отсутствии ошибки либо указывает место ошибки (двоичный порядковый номер ошибочного разряда в слове). Ошибочный разряд автоматически корректируется изменением его состояния на противоположное.

Требуемое число контрольных разрядов (разрядность корректирующего числа) определяется из следующих соображений.

Пусть кодовое слово длиной n разрядов имеет m информационных и k=n-m контрольных разрядов. Корректирующее число длиной k разрядов описывает  $2^k$  состояний, соответствующих отсутствию ошибки и появлению ошибки в первом разряде.

Таким образом, должно соблюдаться соотношение:

$$2^{k} = n - 1$$
 или  $2^{k} - k + 1 = m$ 

**Пример.** Пять контрольных разрядов позволяют передавать в коде Хэмминга до 26 информационных разрядов. Если в ОП одновременно записываются или считываются восемь информационных байт (64 разряда), то при использовании кода Хэмминга потребуется 7 дополнительных контрольных разрядов.

# 2.6.4 Модифицированный код Хэмминга

К контрольным разрядам Хэмминга добавляется еще одни разряд контроля четности всех одновременно считываемых (записываемых) информационных и контрольных разрядов. Модифицированный код Хэмминга позволяет устранять одиночные и обнаруживать двойные ошибки. При использовании в ОП модифицированного кода Хэмминга может производиться коррекция двойных ошибок.

Пусть X — слово, записанное в ОП, а X' — считанное из ОП слово, в котором обнаружены две ошибки. Тогда по сигналу схемы контроля инициируется следующая процедура:

- в неисправную ячейку ОП записывается обратный код считанного слова  $\overline{X}^{'}$  и затем производится его считывание;
- над получаемым при этом кодом ( $\overline{X'}$ ) и кодом  $\overline{X'}$  производится операция сложения по модулю 2.

Полученный код Z содержит 1 в разрядах, в которых имеются ошибки. Схемы управления ОП по коду Z корректируют одну ошибку. После этого схема коррекции одной ошибки исправляет вторую ошибку.

**Пример**. Построение кода Хэмминга с кодовым расстоянием d=3.

Пусть число передаваемых сообщений равно 16, тогда число информационных элементов  $k = \log_2 16 = 4$ . Если выписать все 16 возможных комбинаций, то получим исходный (простой) код. Возьмем только 4 кодовых комбинации в виде единичной матрицы:

Каждую кодовую комбинацию дополним справа проверочными элементами так, чтобы d=3. Нулевая комбинация относится к числу разрешенных комбинаций корректирующего кода, следовательно в проверочные элементы надо добавить не менее двух единиц:

Сложим первые две строки матрицы (2) по модулю 2:

100011 + <u>010011</u> 110000

Так как отличие только в двух элементах, то d=3 не обеспечено. Добавим проверочные элементы следующим образом (можно и в другом виде):

Получили порождающую матрицу кода G(7,4), содержащего 7 элементов, 4 из которых — информационные. Суммируя в различном сочетании строки матрицы (3), получим все (за исключением нулевой) комбинации корректирующего кода с кодовым расстоянием 3. Код G(7,4) обнаруживает двукратные ошибки, а исправляет однократные.

Пусть

 $a_1 - a_7$  — элементы корректирующего кода,

 $a_1 - a_4$  — информационные разряды,

 $a_5 - a_7$  — проверочные разряды.

**Правило формирования проверочного элемента**  $a_j$  (верно для любой кодовой комбинации): Проверочные разряды могут быть получены суммированием по модулю 2 определенных информационных элементов кода. Смотрим столбец  $a_j$  в проверочных разрядах. Если элемент  $a_{ij} = 1$ , то в соответствующей строке і в суммировании участвует информационный элемент, равный 1.

## Пример.

По матрице (3) сформируем элементы  $a_5 - a_7$ .

$$a_5 = a_1 \oplus a_2 \oplus a_4$$
  
 $a_6 = a_1 \oplus a_3 \oplus a_4$  (\*)  
 $a_7 = a_1 \oplus a_2 \oplus a_3$ 

Единицы в позициях информационных разрядов указывают, какие элементы участвуют в формировании проверочного элемента. Единицы в позициях проверочных элементов указывают, какой проверочный элемент (номер) формируется.

Пусть передана кодовая комбинация 100011101, а принята 110011101.

Подставим в проверочные выражения (\*)

$$a_5$$
 = 1  $\oplus$  1  $\oplus$  0 = 0  $\neq$   $a_5$   
 $a_6$  = 1  $\oplus$  0  $\oplus$  0 = 1 =  $a_6$   
 $a_7$  = 1  $\oplus$  1  $\oplus$  0 = 0  $\neq$   $a_7$ 

Синдром (проверочный вектор) определяется комбинацией  $s_1 s_2 s_3$ :

$$s_1 = a_5 \oplus a_5 = 1$$
  
 $s_2 = a_6 \oplus a_6 = 0$   
 $s_3 = a_7 \oplus a_7 = 1$ 

Синдром состоит из нулей, если нет ошибок в передаче, либо ошибки превратили передаваемую кодовую комбинацию в другую разрешенную комбинацию (что маловероятно, но не исключено). Вид синдрома зависит от местоположения одиночной ошибки.

№ ошибочного	1	2	3	4	5	6	7
элемента							
Вид синдрома	111	101	011	110	100	010	001

Таким образом, синдром 101 указывает на ошибку во втором разряде, для исправления которой элемент  $a_2$  меняется на противоположный.

## \*3 СЖАТИЕ ИНФОРМАЦИИ

Цель сжатия — уменьшение количества бит, необходимых для хранения или передачи заданной информации, что дает возможность передавать сообщения более быстро и хранить более экономно и оперативно. Операция извлечения информации с устройства ее хранения будет проходить быстрее, если скорость распаковки данных выше скорости считывания данных с носителя информации.

#### 3.1 ПРОСТЕЙШИЕ АЛГОРИТМЫ СЖАТИЯ ИНФОРМАЦИИ

#### 3.1.1 Метод Шеннона-Фано

Значения дискретной случайной величины (ДСВ) располагают в порядке убывания их вероятностей, а затем последовательно делят на две части с приблизительно равными вероятностями, к коду первой части добавляют справа 0, а к коду второй — 1 слева.

## 3.1.2 Метод Хаффмана

Метод более практичен и никогда по степени сжатия не уступает методу Шеннона-Фано. Код строится при помощи двоичного (бинарного) дерева. Вероятности значений ДСВ приписываются его листьям. Величина, приписанная к узлу дерева, называется весом узла. Два листа с наименьшими весами создают родительский узел с весом, равным сумме их весов. В дальнейшем этот узел учитывается наравне с оставшимися листьями, а образовавшие его узлы от такого рассмотрения устраняются. После постройки корня нужно приписать каждой из ветвей, исходящих из родительских узлов, значения 0 или 1. Код каждого значения ДСВ — это число, получаемое при обходе ветвей от корня к листу, соответствующему данному значению.

Для методов Хаффмана и Шеннона-Фано вместе с собственно сообщением нужно передавать и таблицу кодов.

## 3.1.3 Арифметическое кодирование

Является одной из лучших схем, которая позволяет кодировать некоторые символы менее чем одним битом. По исходному распределению вероятностей для выбранной ДСВ строится таблица, состоящая из пересекающихся только в граничных точках отрезков для каждого из значений этой ДСВ. Объединение этих отрезков должно образовывать отрезок [0,1], а их длины должны быть пропорциональны вероятностям соответствующих значений ДСВ. Алгоритм кодирования за-

ключается в построении отрезка, однозначно определяющего данную последовательность значений ДСВ. Затем для построенного отрезка находится число, принадлежащее его внутренней части и равное целому числу, деленному на минимально возможную положительную целую степень двойки. Это число и будет кодом.

Отрезки строятся так. Если имеется отрезок для сообщения длины n-1, то для построения отрезка для сообщения длины n, разбиваем его на столько же частей, сколько значений имеет рассматриваемая ДСВ. Затем выбирается из полученных отрезков тот, который соответствует заданной конкретной последовательности длины n.

## Алгоритм получения исходного сообщения из его арифметического кода

Шаг 1. В таблице для кодирования значений ДСВ определяется интервал, содержащий текущий код. По этому интервалу однозначно определяется один символ исходного сообщения. Если этот символ — маркер конца сообщения, то конец.

Шаг 2. Из текущего кода вычитается нижняя граница содержащего его интервала, полученная разность делится на длину этого же интервала. Полученное число считается новым текущим значением кода. Переход к шагу 1.

Методы Шеннона-Фано, Хаффмана и арифметическое кодирование относятся к статистическим методам.

# 3.1.4 Словарные алгоритмы

Позволяют кодировать последовательности символов разной длины. Основная идея алгоритма LZ77 состоит в том, что второе и последующие вхождения некоторой строки символов в сообщении заменяются ссылками на ее первое вхождение. Уже просмотренная часть сообщения используется как словарь. Чтобы добиться сжатия, очередной фрагмент сообщения заменяется на указатель в содержимое словаря. Используется "скользящее" по сообщению окно, разделенное на две неравные части. Первая, большая по размеру, включает уже просмотренную часть сообщения. Вторая, намного меньшая, является буфером, содержащим еще незакодированные символы входного потока. Алгоритм пытается найти в словаре фрагмент, совпадающий с содержимым буфера. Обычно размер окна составляет несколько килобайт, а размер буфера — не более 100 байт.

Алгоритм LZ77 выдает коды, состоящие из трех элементов:

1. Смещение в словаре относительно его начала подстроки, совпадающей с началом содержимого буфера.

- 2. Длина этой подстроки.
- 3. Первый символ буфера, следующий за подстрокой.

Пример. Размер окна — 20 символов, в том числе словаря – буфера — 8. Кодируется сообщение "ПРОГРАММНЫЕ ПРОДУКТЫ ФИРМЫ MICROSOFT". Пусть словарь уже заполнен. Тогда он содержит строку "ПРОГРАММНЫЕ ", а буфер — строку "ПРОДУКТЫ". Просматривая словарь, алгоритм обнаружит, что совпадающей подстрокой будет "ПРО", в словаре она расположена со смещением 0 и имеет длину 3 символа, а следующим символом в буфере является "Д". Таким образом, выходным кодом будет тройка <0,3,'Д'>. После этого алгоритм сдвигает влево все содержимое окна на длину совпадающей подстроки +1 и одновременно считывает столько же символов из входного потока в буфер. Получаем в словаре строку "РАММНЫЕ ПРОД", в буфере — "УКТЫ ФИР". В данной ситуации совпадающей подстроки обнаружить не удается, и алгоритм выдаст код <0,0,'У'>, после чего сдвинет окно на один символ. Затем словарь будет содержать "АММНЫЕ ПРОДУ", а буфер — "КТЫ ФИРМ".

Недостатки LZ77:

с ростом размеров словаря скорость работы алгоритма-кодера пропорционально замедляется;

кодирование одиночных символов очень неэффективно.

#### 3.2 ОСОБЕННОСТИ ПРОГРАММ-АРХИВАТОРОВ

Если коды алгоритмов типа LZ передать для кодирования алгоритму Хаффмана или арифметическому, то полученный двухшаговый (конвейерный) алгоритм даст результаты сжатия, подобные широко известным программам: GZIP, ARJ, PKZIP.

Наибольшую степень сжатия дают *двухпроходные* алгоритмы, которые исходные данные последовательно сжимают два раза, но они работают до двух раз медленнее однопроходных при незначительном увеличении степени сжатия.

Большинство программ-архиваторов сжимает каждый файл по отдельности, но некоторые сжимают файлы в общем потоке, что дает увеличение степени сжатия, но одновременно усложняет способы работы с полученным архивом.

Примером программы, имеющей возможность сжимать файлы в общем потоке, является *RAR*. Архиваторы ОС Unix (gzip, bzip2, ...) сжимают файлы в общем потоке практически всегда.

Практически все форматы файлов для хранения графической информации используют сжатие данных.

Сжатие *RLE* (*Run Length Encoding* — кодирование переменной длины) — это простейший метод сжатия, в общем случае очень неэффективный, но дающий неплохие результаты на типичной графической информации. Оно основано на выделении специального кодамаркера, указывающего сколько раз повторить следующий байт.

#### 3.3 Сжатие информации с потерями

Все ранее рассмотренные алгоритмы сжатия информации обеспечивали возможность полного восстановления исходных данных. Но иногда для повышения степени сжатия можно отбрасывать часть исходной информации, т.е. производить сжатие с потерями.

Сжатие с потерями используется в основном для трех видов данных: полноцветная графика ( $2^{24} \approx 16$  млн. цветов), звук и видеоинформация. На первом этапе сжатия с потерями исходная информация приводится (с потерями) к виду, в котором ее можно эффективно сжимать алгоритмами 2-го этапа сжатия без потерь.

#### 3.3.1 Сжатие графической информации

Каждая точка в картинке характеризуется тремя равноважными атрибутами: яркостью, цветом и насыщенностью. Но глаз человека воспринимает эти атрибуты не как равные. Глаз воспринимает полностью только информацию о яркости и в гораздо меньшей степени о цвете и насыщенности, что позволяет отбрасывать часть информации о двух последних атрибутах без потери качества изображения.

Для сжатия графической информации с потерями установлен единый стандарт — формат *JPEG* (*Joint Photographic Experts Group* — название объединения его разработчиков). В этом формате можно регулировать степень сжатия, задавая степень потери качества.

# 3.3.2 Сжатие видеоинформации

При переходе от одного кадра фильма к другому на экране обычно почти ничего не меняется. Таким образом, сжатая видеоинформация представляет собой запись некоторых базовых кадров и последовательности изменений в них. При этом часть информации может отбрасываться. Сжатую подобным образом информацию можно далее сжимать и другими методами. Наиболее распространенными являются стандарты MPEG (Motion Picture Experts Group).

## 3.3.3 Сжатие звуковой информации

Наиболее широко используемый стандарт — это MPEG без видеоданных. Стандарт LPC (Linear Predictive Coding) используется для сжатия речи.

#### Вопросы для самоконтроля

- 1. В чем сущность асинхронной передачи?
- 2. Какие существуют помехи в каналах связи?
- 3. Как бороться с шумом в каналах связи?
- 4. Какие ошибки обнаруживают при контроле по четности?
- 5. В чем смысл коммутации в сетях?
- 6. Перечислить основные характеристики кодов?
- 7. Перечислить методы кодирования информации.
- 8. Назвать коды для помехоустойчивого кодирования?
- 9. В чем заключается принцип блочного кодирования?
- 10. Что означает сжатие информации с потерями?

# Тема 6 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

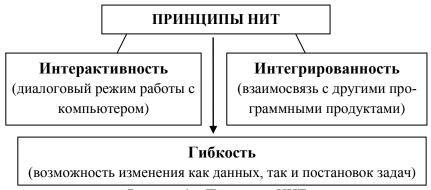
**Информационная технология** — процесс сбора, обработки и передачи данных (первичной информации) для получения информационного продукта.

Новая информационная технология — информационная технология с «дружественным» интерфейсом работы пользователя, использующая персональные компьютеры и телекоммуникационные средства.

Инструментарий информационной технологии — один или несколько взаимосвязанных программных продуктов, технология работы с которыми позволяет достичь поставленную пользователем цель.

Информация является одним из ценнейших ресурсов общества. Процесс ее переработки по аналогии с процессами переработки материальных ресурсов можно воспринимать как технологию.

Цель информационной технологии — производство информации для ее анализа и принятия на его основе решения по выполнению какого-либо действия. Принципы новой информационной технологии (НИТ) приведены на рисунке 1.



# Рисунок 1 – Принципы НИТ

В современном обществе основным техническим средством технологии переработки информации служит персональный компьютер. В качестве инструментария используются: текстовый процессор, настольные издательские системы, электронные таблицы, системы управления базами данных, электронные записные книжки, электронные календари, экспертные системы и т. д.

# 1 КОМПЬЮТЕРНЫЕ ОФИСНЫЕ ТЕХНОЛОГИИ

#### 1.1 ТЕКСТОВЫЙ ПРОЦЕССОР

**Мультиоконность текстового процессора** — способность одновременно работать с несколькими документами, находящимися в различных окнах.

**Абзац** — фрагмент текста, ввод которого закончился нажатием на клавишу Enter.

**Фрагмент** — непрерывная часть текста; может быть строчным, блочным или линейным.

**Форматирование текста** — процедура оформления страницы текста, которая включает:

**Колонтитул** — одна или несколько строк, помещаемых в начале/конце каждой страницы документа; обычно содержит номера страниц, название глав и параграфов.

**Стиль** - это поименованный и сохранённый набор значений параметров форматирования элементов текста.

#### 1.1.1 Работа с документом

Различают логическую и физическую страницы. *Физическая страница* имеет некоторый стандартный размер (например, 210х297 мм). *Логическая страница* образуется на поле физической за вычетом установленных пользователем полей.

*Мягкий переход* на новую страницу осуществляется автоматически после заполнения последней строки на текущей странице.

Жесткий переход (Вставка, Разрыв...) позволяет начать новую страницу, не дожидаясь ее окончательного заполнения.

Висячей строкой называется первая строка или заголовок нового абзаца, оказавшиеся на последней строке страницы, или последняя строка абзаца, оказавшаяся в начале страницы.

*Нумерация страниц* предполагает задание месторасположения на листе номера страницы, отказ от нумерации первой страницы, использование колонтитулов.

Колонтитулы могут различаться для четных и нечетных страниц, для первой страницы и последующих. Использование колонтитулов позволяет лучше ориентироваться в документе, а также использовать дополнительные возможности рекламы.

Работа с документом предполагает выполнение некоторого набора операций. Основной набор операций приведен на рисунке 2.

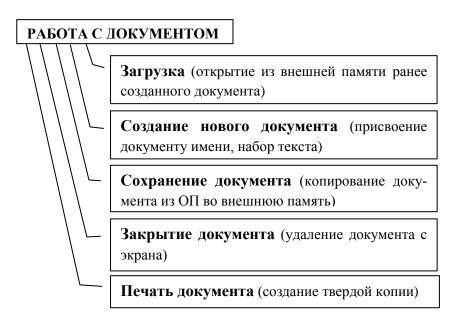


Рисунок 2 – Основной набор операций для работы с документом

Проверка правописания слов и синтаксиса. Наиболее мощные программы проверяют не только правописание, но и склонение, спряжение, пунктуацию и стиль. Указанный режим используют для контроля одного слова, страницы или целого документа.

Контекстный поиск и замена реализуется поиском в документе некоторого текстового элемента и заменой его на заданный пользователем элемент. Элементы могут быть различной длины, включать в себя одно слово, группу слов, часть слова и т.п.

Дополнительные условия для поиска и замены:

Разовая и глобальная замена. Режим разовой замены требует подтверждения пользователя на замену после нахождения каждого вхождения заменяемого элемента. Глобальная замена заменяет все вхождения заменяемого элемента.

Учет регистра символов. Если поисковый элемент "bit", то в режиме учета регистра текстовый процессор пропустит встретившееся ему в тексте слово ВІТ.

*Направление поиска*. Можно реализовать поиск, начиная от позиции курсора в тексте, и до конца документа, либо до начала документа, либо во всем тексте.

Шаблон используется в целях стандартизации документов, которая предполагает унификацию их структуры и внешнего вида. С помощью шаблонов можно составлять бланки различных документов (писем, доверенностей, квитанций и т.п.). Шаблон является типом файла для многократного использования, так как при создании документа на основе шаблона с него автоматически снимается копия, которая и заполняется конкретной информацией.

#### 1.1.2 Форматирование документа

При форматировании выполняются действия, приведенные на рисунке 3.



Рисунок 3 – Набор операций для форматирования документа

Разреженность строк измеряется в *пунктах* (1 пункт = 1/72 дюйма = 0.376 мм). При выборе разреженности учитывается размер шрифта. Низкая разреженность строк затрудняет чтение, а высокая — снижает информативность документа.

Выбор шрифта должен соответствовать передаваемой информации. Различные части документа (заголовок, подзаголовок, текст, примечание) набираются шрифтами разных размеров. Ширина строки (колонки) также зависит от размера шрифта: чем мельче шрифт, тем короче должна быть строка. Тип шрифта (гарнитура) определяет особенности написания символов (заглавные и строчные буквы, цифры, знаки препинания). Каждый шрифт имеет несколько стилей начертания (полужирный, курсив, обычный). Размер шрифта (кегль) по вертикали измеряется в пунктах.

Форматирование может быть применено к фрагменту документа (рисунок 4).



Рисунок 4 – Разновидности фрагментов документа

Перенос. При выключенном режиме автоматического переноса слово, не поместившееся на строке, полностью переносится на следующую строку; при включенном — не поместившееся слово переносится через дефис.

Для форматирования фрагмента текста или абзаца в целом используются различные *стили*. Такими элементами могут быть: заголовок, подпись под рисунком, основной текст и т. д. Стиль позволяет применить к абзацу или отдельному слову несколько атрибутов форматирования за одно действие.

Существуют два типа стилей:

- 1. Стили оформления абзаца. Включают все элементы форматирования, относящиеся к абзацу. По умолчанию используется стиль с названием «Обычный».
- 2. Стили оформления символов. Включают все элементы форматирования влияющие на внешний вид символа: гарнитуру, размер шрифта, его начертание и т.д.

#### 1.2 ЭЛЕКТРОННАЯ ТАБЛИЦА

Электронная таблица — компьютерный эквивалент обычной таблицы, в клетках (ячейках) которой записаны данные различных типов: текст, даты, формулы, числа.

**Рабочая книга** — документ, содержащий несколько листов, в которые могут входить таблицы, диаграммы или макросы.

**Рабочее поле** — пространство электронной таблицы, сетка, в которой создается документ (книга).

Ссылка - способ (формат) указания адреса ячейки.

**Абсолютная ссылка** — это не изменяющийся при копировании и перемещении формулы адрес ячейки, содержащий исходное данное (операнд).

**Относительная ссылка** — это изменяющийся при копировании и перемещении формулы адрес ячейки, содержащей исходное данное (операнд).

**Форматирование данных** — выбор формы представления числовых или символьных данных в ячейке.

Главное достоинство электронной таблицы — возможность мгновенного пересчета всех данных, связанных формульными зависимостями при изменении значения любого операнда. Режимы работы табличного процессора приведены на рисунке 5.

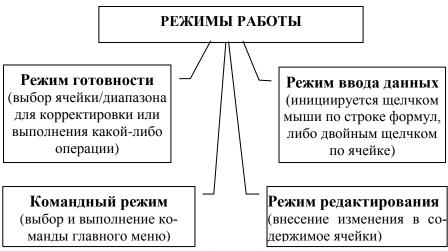


Рисунок 5 – Режимы работы табличного процессора.

Строка формул отображает вводимые в ячейку данные. В ней можно просматривать или редактировать содержимое текущей ячейки. Особенность строки ввода — возможность видеть содержащуюся в текущей ячейке формулу или функцию. При переходе в режим ввода данных прежнее содержимое текущей ячейки теряется.

Рабочая область электронной таблицы состоит из строк и столбцов, имеющих свои имена. Имена строк — это их номера. Нумерация строк начинается с 1 и заканчивается максимальным числом для данного приложения. Имена столбцов — буквы латинского алфавита от A до Z, затем от AA до AZ, BA до BZ и т.д. до IV, либо номера для типа ссылок R1C1. Максимальное количество строк и столбцов опре-

деляется особенностями используемой программы и объемом памяти компьютера. Современные программы дают возможность создавать электронные таблицы, содержащие более 1 млн. ячеек, хотя практически в большинстве случаев этого не требуется.

Пересечение строки и столбца образует *ячейку таблицы*, имеющую свой уникальный адрес. Для указания адресов ячеек в формулах используются *ссылки*.

Текущей (активной) называется ячейка электронной таблицы, в которой в данный момент находится курсор. Существует понятие блока (диапазона) ячеек. В качестве блока ячеек может рассматриваться строка (часть строки), столбец (часть столбца), а также прямоугольник, состоящий из нескольких строк и столбцов или их частей.

## **1.2.1** Формулы

Вычисления в таблицах производятся с помощью формул. Результат вычисления помещается в ячейку, в которой находится формула. Формула может начинаться со знака "равно" и представляет собой совокупность математических операндов, чисел, ссылок, функций. При вычислениях с помощью формул соблюдается принятый в математике порядок выполнения арифметических операций.

Формулы состоят из *операторов* и *операндов*, расположенных в определенном порядке. В качестве операндов используются данные, а также ссылки отдельных ячеек или блоков ячеек, в которых эти данные расположены. Операторы в формулах обозначают действия, производимые с операндами. В зависимости от используемых операторов различают арифметические (алгебраические) и логические формулы. В арифметических формулах используются: операторы (+) сложение, (-) вычитание, (\*) умножение, (/) деление, (^) возведение в степень.

Операторы сравнения: равно (=), неравно (<>), больше (>), меньше (<), не более (<=), не менее (>=). Результатом вычислений арифметической формулы является число.

Логические формулы могут содержать указанные операторы сравнения, а также специальные логические операторы #NOT# — логическое отрицание «НЕ», #AND# — логическое «И», #OR# — логическое «ИЛИ». Логические формулы определяют, выражение истинно или ложно. Истинным выражениям присваивается численная величина 1, а ложным — 0.

По умолчанию электронная таблица вычисляет формулы при их вводе и пересчитывает их повторно при каждом изменении входящих в них исходных данных.

#### 1.2.2 Функции

 $\Phi$ ункции можно считать частным случаем формулы. Функция представляет собой программу с уникальным именем, для которой пользователь должен задать конкретные значения аргументов, стоящих в скобках после ее имени. Под функцией понимают зависимость одной переменной (у) от одной (х) или нескольких переменных ( $X_1$ ,  $X_2$ , ...,  $X_n$ ). Функции вводят в таблицу в составе формул или отдельно.

Математические функции выполняют различные математические операции (вычисление логарифмов, тригонометрических функций, преобразование радиан в градусы и т.п.).

Статистические функции выполняют операции по вычислению параметров случайных величин или их распределений, представленных множеством чисел (стандартное отклонение, среднее значение, медиана и т. п.).

Текстовые функции выполняют операции над текстовыми строками или последовательностью символов, вычисляя длину строки, преобразуя заглавные буквы в строчные и т. п.

Финансовые функции используются в сложных финансовых расчетах (определение амортизационных отчислений и др.)

Даты имеют внутренний (количество дней от начала 1900 года) и внешний формат. Этот тип данных обеспечивает выполнение таких функций, как добавление к дате числа (пересчет даты вперед/назад) или вычисление разности двух дат (длительность периода). Внешний формат используется для ввода и отображения дат.

Примеры типы внешних форматов дат

ДД-МММ-ГГ(04-Янв-95)

МММ-ДД-ГГ(Янв-04-95)

ДД-МММ(04-Янв)

MMM- $\Gamma\Gamma$  (Янв-95)

## 1.2.3 Типы и форматы данных

Тип входных данных, содержащихся в каждой ячейке, определяется первым символом, который должен трактоваться не как часть данных, а как команда переключения режима:

• если в ячейке содержатся числа, то первый их символ является либо цифрой, либо десятичной точкой, либо знаком числа (плюсом или минусом);

- если в ячейке содержится формула, то часто используются левая круглая скобка, знак числа (плюс или минус), знак равенства и т. п.;
- если ячейка содержит символьные данные, ее первым символом может быть одинарная (апостроф) или двойная кавычка, или пробел.

Символьные (текстовые) данные могут включать в себя алфавитные, числовые и специальные символы.

*Числовые данные* не могут содержать алфавитных и специальных символов, поскольку с ними производятся математические операции. Единственными исключениями являются десятичная точка (запятая) и знак числа, стоящий перед ним.

Основной формат используется по умолчанию для записи данных в ячейках в том виде, как они вводятся или вычисляются.

Формат с фиксированным количеством десятичных знаков обеспечивает представление чисел с заданной точностью, которая определяется установленным пользователем количеством десятичных знаков после запятой (десятичной точки).

**Пример.** Пусть установлен режим форматирования 2 десятичных знака. Тогда вводимое в ячейку число 12345 будет записано как 12 345.00; число 0.12345 — как 0.12.

Процентный формат обеспечивает представление данных в форме процентов со знаком % (в соответствии с установленным количеством десятичных знаков)

**Пример.** Если установлена точность в один десятичный знак, то при вводе 0.123 на экране появится 12.3 %, а при вводе 123 — 12300.0 %.

Денежный формам обеспечивает представление чисел, при котором каждые три разряда разделены запятой; пользователь может установить определенную точность представления (округление).

**Пример.** Введенное число 12345 будет записано в ячейке как 12 345 (с округлением до целого числа) и 12 345.00 (с точностью до двух десятичных знаков).

Научный (экспоненциальный) формат используется для представления очень больших или очень маленьких чисел, в виде двух компонентов:

• *мантиссы*, имеющей один десятичный разряд слева от десятичной точки, и некоторого количества десятичных знаков справа от нее;

#### • порядка числа.

Запись E±n означает умножение мантиссы на 10 в степени ±n.

**Пример.** Число 12345 будет записано в ячейке как 1.2345E+04 (при точности 4 разряда) и как 1.23E+04 (при точности в 2 разряда). Число .0000012 будет иметь вид 1.2E-06.

Если ширина вводимого числа превышает ширину ячейки (столбца), то ячейка заполняется символами ###. Это сигнал о том, что ширина ячейки недостаточна для отображения данных.

## 1.2.4 Типы адресации

Адрес ячейки определяется названием столбца и номером строки. Адрес и содержимое текущей ячейки выводятся в строке ввода электронной таблицы. Адрес блока ячеек задается указанием ссылок первой и последней его ячеек, между которыми ставится разделительный символ двоеточие <:> или две точки подряд <..>.

Для указания абсолютной адресации вводится символ \$. Различают два типа абсолютной ссылки: полная и частичная.

Полная абсолютная ссылка указывается, если при копировании или перемещении адрес ячейки, содержащей исходное данное, не меняется. Для этого символ \$ ставится перед наименованием столбца и номером строки, например, \$A\$1.

*Частичная абсолютная ссылка (смешанная)* указывается, если при копировании и перемещении не меняется строка или столбец.

**Пример 1.** Ссылка формата А\$1 является относительной по столбцу и абсолютной по строке. При копировании ячейки с формулой выше/ниже текущего положения ссылка изменяться не будет. При копировании влево/вправо будет изменяться адрес столбца.

**Пример 2.** Ссылка формата \$A1 является абсолютной по столбцу и относительной по строке. При копировании ячейки с формулой выше/ниже текущего положения будет изменяться адрес строки. При копировании влево/вправо ссылка изменяться не будет.

В *относительной ссылке* на ячейку используют имя столбца и имя строки, на пересечении которых она расположена, например, A1.

## 1.2.5 Графические возможности

Использование графики повышает наглядность полученных результатов, показывает соотношение различных значений и динамику их изменения. Например, *круговая диаграмма* применяется для сравнения отдельных значений переменной между собой и с общей их суммой; *вертикальная столбцовая* используется для сравнения значе-

ний переменных в различные моменты времени; линейный график используется для иллюстрации динамики переменной во времени.

## 1.3 Система управления базами данных

**База данных** (БД) — совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимо от прикладных программ.

Система управления базами данных (СУБД) представляет собой комплекс программ и языковых средств для создания, ведения и использования БД.

**Запись** — это набор взаимосвязанных данных о конкретном предмете.

Поле - информационная единица записи.

**Ключ** (ключевое поле) — поле, которое содержит наиболее характерную для данной записи информацию.

**Группа** — это идущие подряд записи с одинаковым одним или несколькими полями.

Для эффективной работы с базой данных СУБД должна обеспечивать целостность и непротиворечивость данных. Это физическая сохранность данных, предотвращение неверного их использования, защита от искажений и несанкционированного доступа.

Основная особенность СУБД — наличие процедур для ввода и хранения не только самих данных, но и описаний их структуры. СУБД предоставляет возможности выбирать любые поля, форматы полей, сортировать данные, вычислять итоговые значения, отбирать данные по какому-либо признаку, менять их, удалять, копировать в другие таблицы и т.д. Основные функции СУБД приведены на рисунке 6.



Запрос позволяет указать, какие именно поля будут отражаться; ввести критерий (правило), согласно которому будут определяться

отображаемые записи и порядок сортировки этих записей. Запросы можно использовать в качестве основы для форм или отчетов, чтобы определить, какие именно записи должны в них содержаться.

 $\Phi$ ормы позволяют управлять процессом отображения данных на экране.

Для вывода на печать используются *отчеты*, которые можно отображать в режиме *Конструктора*, либо в окне предварительного просмотра.

Объекты базы данных приведены на рисунке 7.



Рисунок 7 – Объекты базы данных

Основными объектами базы данных являются *таблицы*. Данные, которые они содержат, являются основой любой СУБД. Организация данных напоминает лист книги электронной таблицы. В каждом столбце таблицы находятся данные одинакового типа, а каждая строка содержит одинаковые записи.

Критерии оценки производительности СУБД приведены на рисунке 8.

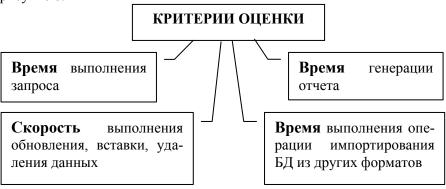


Рисунок 8 – Критерии оценки производительности СУБД

#### 1.3.1 Модели данных

Модели данных различаются по способу установления связей между данными (рисунок 9).



Рисунок 9 – Модели данных

Реляционная («relation» — отношение) модель является простейшей и наиболее привычной формой представления данных в виде таблицы. Достоинством реляционной модели является сравнительная простота инструментальных средств поддержки, недостатком — жесткость структуры данных и зависимость скорости работы от размера базы данных.

Иерархическая и сетевая модели предполагают наличие связей между данными, имеющими какой-либо общий признак. В иерархической модели такие связи могут быть отображены в виде дереваграфа, где возможны только односторонние связи от старших вершин к младшим. В сетевой модели теоретически возможны связи "всех со всеми". Использование иерархической и сетевой моделей ускоряет доступ к информации в базе данных. Но поскольку каждый элемент данных должен содержать ссылки на некоторые другие элементы, требуются значительные ресурсы как дисковой, так и оперативной памяти персонального компьютера.

**Пример.** Классификация животного мира, структура почтовых адресов, файловая структура диска относятся к иерархической модели баз данных.

## 1.3.2 Типы и форматы данных

Типы данных: *Текстовый, Числовой, Дата/Время, Денежный, Счетчик.* Значения числовых полей по умолчанию до ввода данных равны нулю.

Числовой формат	Диапазон	Значение
Целое	-32768 +32767	Целое

Форматы Дата/Время	Отображение
Полный формат даты	21.07.03 13:05:41
Длинный формат даты	21 июля 2003г.
Средний формат даты	21-июл-03
Краткий формат даты	21.07.03
Длинный формат времени	13:05:41
Средний формат времени	01:05
Краткий формат времени	13:05

#### 1.3.3 Поля таблины

В реляционных БД отдельные таблицы связаны между собой по ключевым полям. Обязательное условие установления связи между таблицами — наличие двух одинаковых полей. Поле, с помощью которого осуществляется связывание таблиц, в главной (родительской таблице) называется главным (первичным), а в подчиненной (дочерней) — внешним. Поля, являющиеся главными и внешними ключами, должны иметь одинаковые имя, тип и размер. В родительской таблице значения главного ключа должны быть уникальны. В дочерней таблице значения внешнего ключа могут повторяться. Поле первичного ключа НЕ позволяет вводить в таблицу повторяющиеся записи, поскольку содержит однозначный идентификатор для каждой записи.

Свойства полей	Описание	
Размер	Максимальная длина текстового или число-	
	вого поля (если задан слишком большой	
	размер, то неэкономно расходуется память)	
Формат	Форма отображения данных:	
	С разделителями разрядов (пробел между	
	триадами, запятая между целой и дробной	
	частью и 2 знака после запятой);	
	Основной (число отображается так, как его	
	ввел пользователь).	
Подпись	Выводится рядом с полем в форме или от-	
	чете, содержит пояснение	

Значение по умолчанию	Пример. Если большинство студентов		
	имеют оценку «5», то это значение можно		
	установить по умолчанию для всех студен-		
	тов. В записях для студентов, имеющих		
	другие значения оценки, выполняется кор-		
	ректировка.		
Условие на значение	Устанавливает ограничения на вводимые		
	значения (например, муж/жен)		
Сообщение об ошибке	Появляется при вводе недопустимого зна-		
	чения.		
Обязательное поле	Поле, подлежащее обязательному заполне-		
	нию.		

Функция поиска позволяет перейти к полю, содержимое которого известно. Символы \* и ? выполняют те же функции, что и при поиске файлов. Поиск с заменой выполняется аналогично.

При удалении записей в поле первичного ключа нумерация не изменяется. Например, если было 5 записей с номерами от 1 до 5, то после удаления третьей и четвертой записей, оставшиеся будут с прежними номерами 1, 2, 5.

Для компактного отображения полей таблицы часть из них может быть скрыта. Скрытые поля не участвуют в поиске и обработке запросов (Скрыть столбцы/Отобразить столбцы).

Для повышения удобочитаемости таблицы используют фиксацию маркированных полей (Закрепить столбцы/Освободить все столбцы). В результате фиксированное поле помещается в левой части таблицы и остается видимым при прокрутке полей вправо.

Уменьшение размера поля, в которое вводится данное, ведет к его обрезке по новой границе.

## 1.3.4 Запросы

Наиболее распространенный запрос — запрос на выборку, который выполняет отбор данных из одной или нескольких таблиц по некоторым критериям (рисунок 10). Конструктор позволяет сконструировать запрос на выборку самостоятельно.



Рисунок 10 – Разработка запроса

Простой запрос создается на основе выбранных полей.

*Перекрестный запрос* предоставляет данные в компактном формате как в электронных таблицах.

При составлении запроса на основе нескольких таблиц между ними должны быть установлены связи, иначе результат обработки запроса может оказаться некорректным.

Критерии отбора, расположенные в разных строках, объединяются логическим оператором ИЛИ; расположенные в одной строке — логическим оператором И.

Условия И выполняются раньше условий ИЛИ.

Для исключения группы данных надо записать критерий HE PABHO «значение» или <> «значение».

В запросах символы \* и ? применяются также как и во всех приложениях Microsoft Office

Для удаления поля из бланка запроса используется клавиша Delete. Для удаления таблицы из запроса надо ее выделить, щелкнув по полю в верхней части окна конструктора запроса и нажать Delete.

Запрос может быть использован для выполнения расчетов и подведения итогов на основе обобщенных данных исходных таблиц.

Над полученными в результате запроса данными можно провести вычисления с помощью функций.

Функция	Назначение	
Группировка	Объединение в одну группу записей с одинаковыми	
	значениями	
Count	Определение числа записей в группе	
Sum	Суммирование значений поля в отобранных запро-	
	сом записях	

Min	Определение минимального значения в заданном по-
	ле группы записей
Max	Определение максимального значения в заданном
	поле группы записей
Avg	Определение среднего значения в заданном поле
-	группы записей
First	Определение первого значения в заданном поле
	группы записей
Last	Определение последнего значения в заданном поле
	группы записей

#### 1.3.5 Фильтры

 $\Phi$ ильтры (рисунок 11) используются в тех же целях, что и запросы на выборку данных. При фильтрации нельзя подавить отображение отдельных полей и выполнить вычисления. В фильтре по выделенному можно выбрать команду  $\Phi$ ильтр, Исключить выделенное.



Рисунок 11 – Разновидности фильтров

#### 1.3.6 Формы

Pежим таблицы используется, если надо получить наиболее полный обзор данных или для эпизодического изменения данных. При частом изменении данных лучше использовать форму.

Формы используются как средство защиты БД от действий неквалифицированных пользователей, повышают комфортность работы. Любая форма строится на основе таблицы или запроса; поля располагаются произвольно. Логически взаимосвязанные поля объединяются в группы. На основе одной таблицы могут быть построены разные формы для различных работников.

*Мастер форм* в зависимости от назначения формы предлагает на выбор стандартные шаблоны и стили оформления.

*Конструктор* предоставляет инструменты для создания формы на пустом бланке.

*Мастер сводных таблиц* создает сводную электронную таблицу на основе таблиц и запросов СУБД.

Мастер диаграмм создает формы с диаграммами на основе выбранных полей таблицы. По умолчанию значения столбца образуют на диаграмме ряд данных, а значения строки — категорию. Для отображения числовых значений служат маркеры данных — линии, полосы, секторы и т.п. Метки у маркеров данных представляют числовые или текстовые значения элементов ряда. Оси используются для отображения шкалы — диапазона значений рядов данных. Легенда — пояснительный текст и свод условных знаков.

Каждый элемент диаграммы имеет набор параметров, который можно изменить в соответствующем диалоговом окне, дважды щелкнув на нужном элементе. *Круговую диаграмму* рекомендуется использовать для отображения соотношения частей целого. *Точечные* диаграммы используют для представления взаимосвязи между двумя или несколькими величинами. *Гистограммы* используют для сравнения значений, относящихся к определенному моменту времени.

Содержимое формы всегда соответствует информации в таблицах и запросах и обновляется при каждом открытии. Удалить запись из формы можно клавишей *Delete*.

#### 1.3.7 Отчеты

Преимуществом отчетов перед формами является возможность группировки данных, что не только улучшает его внешний вид, но и повышает читабельность. Данные в отчете группируются не более чем по трем полям с возможностью изменения уровня группировки. В нижнем колонтитуле отчета автоматически устанавливается дата его создания.

3аголовок расположен на первой странице отчета перед верхним колонтитулом.

Верхний колонтитул по умолчанию помещается на каждую страницу отчета; обычно содержит заголовки колонок. На страницах, содержащих шапку или резюме отображение колонтитула можно заблокировать.

Заголовок группы состоит из имени группы и названия Заголовок группы.

Элементы области данных повторяются для каждой выбранной из таблицы записи.

*Примечание группы* появляется в конце группы записей; применяется для указания числа записей или суммы значений в полях группы.

*Нижний колонтитул* присутствует на каждой странице отчета; используется для отображения номера страницы и текущей даты.

*Примечание отчета* вставляется в конце отчета и содержит резюме ко всему документу.

По умолчанию все области отчета имеют одинаковую ширину. Изменить высоту и ширину области одновременно можно перемещением слева маркировочного квадратика, либо поочередным перемещением ограничительной линии сверху и справа.

Нумерация страниц отчета выполняется в текстовом поле, добавленном в заголовки или колонтитулы.

#### 1.4 РЕДАКТОР ПРЕЗЕНТАЦИЙ

Приложение Impress из пакета Libre Office поддерживает создание презентаций с рисунками, звуковыми эффектами и анимацией; экспорт данных в форматы PDF, PPT, SWF; текстовые и фоновые стили создания презентаций. Impress — упрощённая альтернатива PowerPoint и другому профессиональному ПО для создания презентаций. Программа абсолютно бесплатна, присутствует на всех платформах и совместима даже со старыми версиями операционных систем.

Приложение Prezi. Создатели этого проекта отказались от привычного формата слайдов. Презентация выглядит как одна большая карта, на которой размещается текст, видео, графика и пр. Во время показа изображение движется не от слайда к слайду, а от одного участка карты к другому. Нужные области увеличиваются с помощью красивых эффектов. Prezi лучше всего подходит для креативного представления идей. Есть возможность работать в команде онлайн.

Приложение PowerPoint — редактор презентаций из пакета Microsoft Office. В 2013 году вышла модифицированная версия, посредством которой качественные презентации создаются максимально просто.

Типы слайдов презентации: экран (для показа на экране компьютера или через мультимедийный проектор), лист А4 (для подго-

товки раздаточного материала), прозрачка (для показа через слайдпроектор) и др.

В приложении PowerPoint предусмотрены поддержка видео и звука, анимации; импорт данных (таблиц Excel, диаграмм) из других приложений пакета Microsoft Office; возможности для создания полиграфии и альбомов;

PowerPoint — самая популярная программа, которая позволяет работать непосредственно в сети, редактировать и просматривать презентации, сохранять их в облачном хранилище. Версия программы создания презентаций 2013 года адаптирована для использования приложения в мобильных устройствах, планшетах с сенсорным экраном.

## Основные режимы PowerPoint

Обычный режим. Основной рабочий режим в процессе создания презентации. В нем отображаются три области: структуры, слайда и заметок. Размеры областей можно изменять, перетаскивая их границы. На слайды можно добавлять текстовые блоки, вставлять рисунки, таблицы, организационные диаграммы, графические объекты, надписи, фильмы, звуковые эффекты, гиперссылки и анимацию.

Режим слайда позволяет отобразить текущий слайд в крупномасштабном виде.

Режим структуры. Презентация отображается в сокращенном виде (только заголовки слайдов и содержащийся в слайдах текст). Можно поменять местами пункты на слайде, переместить слайд целиком с одной позиции в другую, изменить заголовки и основной текст. Основной текст расположен после каждого заголовка слайда и включает до пяти уровней отступов.

Режим сортировщика слайдов позволяет просматривать слайды в виде эскизов, что упрощает ее реорганизацию, добавление, удаление слайдов, а также предварительный просмотр эффектов анимации и смены слайдов. Этот режим позволяет оценить вид презентации в целом, насколько единообразно она построена.

Режим показа слайдов используется для предварительного просмотра, репетиции готовой презентации и для ее реального показа. Презентация демонстрируется в том виде, как она предстанет перед аудиторией.

Режим страниц заметок служит для добавления заметок докладчика или сведений для аудитории в раздаточных материалах. Если в заметках должен быть рисунок, то добавлять его следует в режиме страниц заметок.

## 2 СИСТЕМЫ КОМПЬЮТЕРНОЙ ВЕРСТКИ

#### 2.1 Система верстки ТеХ

TeX — это созданная американским математиком и программистом Дональдом Кнутом система для верстки текстов с формулами.

ТеХ не является системой WYSIWYG как текстовый процессор Word. ТеХ представляет собой специализированный язык программирования, на котором пишутся издательские системы. Каждая издательская система на базе TeX'a представляет собой пакет макроопределений (макропакет) этого языка.

LaTeX – это созданная Лесли Лампортом издательская система на базе TeX'a. Plain-TeX создан Дональдом Кнутом для построения более сложных систем; на практике используется как средство для обмена текстами.

AMS-TeX – издательская система, ориентированная на верстку статей для математических журналов и книг, издаваемых Американским математическим обществом.

Современные версии LaTeX'а включают в себя AMS-ТеX'овские возможности для набора формул. В первом приближении можно воспринимать слова TeX и LaTeX как синонимы.

Все издательские системы на базе TeX'а обладают достоинствами, заложенными в самом TeX'е: напечатанный текст выглядит так как в типографской книге. При работе с TeX'ом можно не вникать в детали оформления, указав логическую структуру текста. Для изменения формата достаточно заменить данные в стилевом файле. Огромным достоинством систем на базе TeX'а является высокое качество и гибкость верстки абзацев и математических формул.

TeX'овские файлы обладают высокой степенью переносимости. Поэтому TeX очень популярен как язык международного обмена статьями по математике и физике.

Для создания документа типографского качества нужно подготовить файл с текстом (в любом текстовом редакторе), состоящим из команд для TeX'a/LaTeX'a. Файл с исходны текстом по традиции имеет расширение "tex". Далее tex-файл обрабатывается с помощью программы-транслятора. В результате получается файл с расширением "dvi" — не зависящий от устройства. Полученный dvi-файл можно посмотреть на экране и распечатать на принтере.

Для подготовки публикаций в области естественных наук используют специализированные макропакеты.

#### 2.1.1 Макропакеты для физики

bohr — пакет для изображения атома в простейшей модели Бора.

feyn — пакет для создания простых фейнмановских диаграмм.

feynmf — предоставляет стили feynmf/feynmp, которые содержат набор макросов для создания сложных фейнмановских диаграмм средствами пакетов METAFONT и MetaPost.

*hepnames* и *hepparticles* — два независимых набора макросов для печати названий частиц для физики высоких энергий.

isotope и nuc — пакеты для печати названий изотопов химических элементов.

miller — макрос для печати индексов Миллера.

*physics* и *physymb* — набор мелких полезных макросов для физиков, упрощающих набор документов.

*pst-optic* и *pst-optexp* — создание иллюстраций для геометрической оптики и схем оптических экспериментов.

 $\textit{pst-diffraction} \ -\!\!\!-\!\!\!-\!\!\!-\!\!\!\!-$  подготовка иллюстраций для экспериментов по дифракции.

pst-electricfield — визуализация электрического поля простой системы точечных зарядов.

pst-magneticfield — визуализация магнитного поля.

*pst-pulley* — изображение системы блоков и грузиков.

pst-solarsystem — иллюстрация Солнечной системы.

pst-spectra — изображение спектров излучения и поглощения.

randomwalk — случайно генерируемые пути (имитация Броуновского движения).

SIstyle, SIunits и siunitx — пакеты с единицами физических величин по правилам СИ.

units — печать единиц физических величин.

unitsdef — печать единиц физических величин, в том числе и по правилам СИ.

tqft — создание диаграмм для топологической квантовой теории поля.

venndiagram — создание диаграмм Венна.

## 2.1.2 Макропакеты для информатики

Возможности LaTeX позволяют готовить документацию для готового программного обеспечения или требования для проектов, разработка которых только начинается. Можно визуализировать алго-

ритмы из классической информатики с помощью стилей algorithms, algorithmic и pseudocode и др.

alg — предоставляет два окружения для печати алгоритмов.

dirtree — визуализация дерева файловой системы.

flowchart — создание блок схемы программы, следующей рекомендациям IBM Flowchart Template (ISO 1028:1973).

listings — пакет для оформления программного кода с поддержкой множества языков программирования.

## 2.1.3 Макропакеты для электроники

LaTeX не может рассматриваться в качестве пакета для проектирования электрических схем, но некоторые возможности есть.

bloques — механизм для рисования схем силовой электроники.

*eltex* — электрические схемы с сеткой, упрощающей расположение элементов.

*circ* — создание несложных электрических схем средствами LaTeX и *METAFONT*.

circuitikz — визуализация электрических схем средствами TikZ.

pst-circ — простые электрические схемы.

pst-osci — иллюстрации осциллограмм.

## 2.1.4 Макропакеты для химии

Возможности по представлению химических формул и реакций в LaTeX позволяют решить большинство задач, возникающих при подготовке соответствующих публикаций.

*bpchem* — печать химических элементов, названий, формул и нумерация химических соединений.

*mhchem* — пакет для печати названий химических элементов, формул и значков для маркировки опасных химических соединений.

chemarr из пакета oberdiek — дополнительный набор стрелок для химиков.

*chemarrow* — ещё один дополнительный набор стрелок для химиков.

*chemexec* — набор простейших химических макросов.

*chemfig* — пакет для работы со структурными формулами.

*chemmacros* — обширный набор макросов для химиков. Пакет предоставляет стили chemmacros, chemformula, ghsystem и chemgreek.

снетпит — маркировка химических соединений.

chemcompounds — простая последовательная нумерация химических соединений с возможностью присвоения каждому соединению своего уникального имени.

*chemcono* — нумерация химических соединений по принципу организации библиографических ссылок.

*chemstyle* — набор стилей для оформления публикаций в общепринятом "химическом" стиле.

cryst — набор символов для описания типа симметрии в кристаллографии.

lewis — электронные структуры Льюиса.

*modiagram* — позволяет рисовать молекулярные орбитальные диаграммы.

mychemistry — рисование схем химических реакций.

substances — создание банка данных по химическим веществам.

*хуттех* — пакет для отображения средствами LaTeX химических структурных формул.

#### 2.2 СИСТЕМА ВЕРСТКИ SCRIBUS

Scribus — это инструмент компьютерной графической верстки с открытым исходным кодом. Ориентирован на макет, фреймы, стили, управление изображениями и межколоночное перетекание текста. Предназначен для компьютерной верстки и поэтому использует фреймы как фундаментальную концепцию.

Scribus работает на платформах Linux, Windows и Mac OS X. Генерирует документы в формате PDF, которые можно печатать на стандартных лазерных и струйных принтерах, а также на специальном профессиональном оборудовании растрирования изображений. Поддерживает встроенные команды LaTeX.

Поддержка импорта файлов в форматах csv (текст с разделителями), doc (Microsoft® Office Word), html(язык разметки гипертекста), odt (Open Document Text ), txt (стандартный текст).

# 3 СИСТЕМЫ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ

Компьютерная математика — совокупность теоретических, методических, программных и аппаратных средств, позволяющих производить математические вычисления с высокой степенью точности и производительности. Термин «компьютерная математика» является обобщением различных разделов математики. Речь идет об автоматизации решения математических задач на компьютерах.

СКМ — системы компьютерной математики — универсальный математический инструмент, ориентированный на широкий круг пользователей, которые не являются профессионалами в области математики или программирования. СКМ — это, мощные электронные справочники, базы данных по всем современным направлениям математики, эффективные средства решения математических задач.

#### Классы СКМ

- 1. Системы для численных расчетов.
- 2. Табличные процессоры.
- 3. Матричные системы.
- 4. Системы для статистических расчетов.
- 5. Системы для специальных расчетов.
- 6. Системы для аналитических расчетов.
- 7. Универсальные системы.

Каждая СКМ имеет определенные свойства, которые необходимо учитывать при решении конкретных математических задач.

*MAPLE* — система символьных вычислений или система компьютерной алгебры. Позволяет выполнять как численные, так и аналитические расчеты с возможностью редактирования текста и формул на рабочем листе.

*Mathematica* имеет широкий набор средств, переводящих сложные математические алгоритмы в программы. В память компьютерной системы Mathematica заложены алгоритмы, содержащиеся в курсе высшей математики технического вуза.

*MATLAВ* — язык высокого уровня для научно-технических вычислений.

*МАТНСАD* служит средством вычислений, анализа и написания отчетов для профессионалов во всех областях науки и техники. Система проста в использовании и не вызывает проблем при обучении.

## 4 КОМПЬЮТЕРНАЯ ГРАФИКА

#### 4.1 РАЗНОВИДНОСТИ КОМПЬЮТЕРНОЙ ГРАФИКИ

Компьютерная графика — область информатики, изучающая методы и средства создания и обработки графических изображений.

По способу формирования изображений различают *векторную*, *растровую* и *фрактальную* графики. В *3D-графике* сочетаются растровый и векторный способы формирования изображений.

Векторные данные могут включать в себя данные о типе линии и соглашения о том, как они будут вычерчиваться. Линии используются для построения геометрических фигур. Векторные данные представляют собой список операций черчения и математическое описание элементов изображения, записанные в той последовательности, в которой они создавались. Векторные данные гораздо менее объемные, чем растровые. Простейшая векторная графика используется в текстовых редакторах, электронных таблицах, редакторах презентаций.

Векторные форматы удобны для хранения изображений, которые могут быть разложены на простейшие геометрические объекты. Векторные изображения легко масштабируются, но плохо отображаются на растровых устройствах вывода.

Растровые форматы специально разрабатывались для хранения реальных изображений. Растровые файлы имеют очень большой размер, особенно если изображение многоцветное. Применение различных схем сжатия уменьшает размер. Распаковка сжатого файла перед использованием замедляет процесс визуализации изображения. Растровые изображения плохо поддаются масштабированию.

Началом фрактальной графики считается издание в 1977 г. книги «Фрактальная геометрия в природе» Бенуа Мандельброта. *Фрактал* — это структура, состоящая из частей, которые в каком-то смысле подобны целому. В любом фрактале найдется такая часть, которая содержит информацию о всем фрактале. Фрактальными свойствами обладают многие природные объекты, такие как снежинка, кристаллы, растения.

## 4.2 Цветовые модели

Аддитивная модель. Новые цвета получаются при сложении основного цвета с черным. Чем больше интенсивность добавляемого цвета, тем ближе результирующий цвет к белому. Если при смешивании всех основных цветов значения их интенсивности максимальны

(255), то получим чисто белый цвет. Если интенсивности минимальны (0), то получим черный цвет.

Субтрактивная модель. Для получения всех цветов основные цвета вычитаются из белого. Чем больше интенсивность вычитаемого цвета, тем ближе результирующий цвет к черному. Смешивание всех основных цветов дает черный (значение интенсивности максимально) или белый (значение интенсивности равно нулю).

*Цветовая модель RGB* (Red Green Blue) — аддитивная цветовая модель, основанная на трех цветах: красном, зеленом, голубом. Если все три цвета равны, то это оттенки серого цвета. В модели RGB значения первичных цветов (0,0,0) дают черный цвет, (255,255,255) — белый. Модель RGB используется в мониторах.

Палитра также называется картой индексов, таблицей цветов или таблицей перекодировки и представляет собой одномерный массив цветных величин. Фрагмент палитры.

R	G	В	Цвет
255	0	0	Красный
0	255	0	Зеленый
0	0	255	Синий
255	255	255	Белый
127	127	127	Серый

В больших и сложных изображениях косвенное сохранение цветов с использованием палитры экономит память за счет сокращения объема данных. Если объем растровых данных в файле невелик или используется больше 255 цветов, то включение в файл палитры может привести к обратному результату, т.е. требуемый объем памяти возрастет. Для изменения цвета изображения достаточно изменить в палитре значение одного цвета на значение другого. Палитру неэффективно применять для полноцветного изображения.

*Цветовая модель СМҮ* (Cyan Magenta Yellow — голубой, пурпурный, желтый) — противоположность модели RGB. Это субтрактивная цветовая модель. При освещении каждый из основных цветов поглощает дополняющий его цвет, например, голубой поглощает красный, пурпурный — зеленый, желтый — синий. Теоретически при вычитании всех основных цветов, суммой является черный, но на практике получить черный сложно. Поэтому модель дополнена отдельным черным цветовым компонентом. *Цветовая модель СМҮК* (К от Black — последняя буква). Черный цвет является в этой модели

основным. Результат применения этой модели называется 4-хцветной печатью. Модель используется в принтерах.

*Цветовая модель HSV* (Hue Saturation Value — оттенок, насыщенность, величина). *Насыщенность* представляет собой количество белого, *величина* — количество черного, *оттенок* — тот цвет, к которому добавляется белый и черный. В этой модели при определении новых цветов изменяют их свойства. Оттенок — это цветовая насыщенность (цветность), которая определяет количество белого в оттенке. В полностью насыщенном 100%-ном оттенке не содержится белого. Такой оттенок считается чистым.

*Цветовая модель YUV* основана на линейном преобразовании данных RGB-изображения. Применяется для кодирования цвета в телевидении. У определяет яркость, UV — цветность.

#### 4.3 Сжатие графических изображений

Сжатие — это процесс, применяемый для уменьшения физического размера блока информации. При сжатии программа компрессор осуществляет сжатие данных, а программа декомпрессор – их восстановление.

Метод сжатия без потерь — когда данные сжимается, а потом распаковываются и содержащаяся в них оригинальная информация сохраняется. Данные не должны быть изменены, потеряны или повреждены.

Сжатие с потерями предполагает отбрасывание некоторых данных изображения для достижения лучшей степени сжатия, чем при использовании методов без потерь.

Фрактальное сжатие — это математический процесс, который применяется для кодирования растров, содержащих реальные изображения. В результате получается совокупность математических данных, которые описывают фрактальные свойства изображения. Фрактальное кодирование основано на том, что большинство объектов содержат избыточную информацию в виде одинаковых повторяющихся фрагментов. Фрактальное сжатие — сжатие с потерями, так как процесс сравнения фракталов не предусматривает поиска точного их соответствия. Параметрами сжатия являются время кодирования, качество изображения и размер выходного файла. Высокая степень сжатия достигается путем выполнения большого объема преобразований и вычислений.

В процессе преобразования реальных растровых данных во фрактальные коды реализуется возможность масштабирования фрактальных изображений без введения артефактов и потери деталей. Размер физических данных, используемых для записи фрактальных кодов значительно меньше размеров исходных растровых данных. Степень сжатия реального изображения с помощью фрактального кодирования достигает 200:1.

Требования к алгоритмам сжатия

- 1. степень компрессии;
- 2. скорость компрессии;
- 3. скорость декомпрессии;
- 4. качество изображения;
- 5. масштабирование изображения;
- 6. редактируемость (изменение качества изображения при его повторном сохранении).

#### 4.4 ГРАФИЧЕСКИЕ ФОРМАТЫ

Все множество форматов, используемых для записи изображений, можно условно разделить на три категории:

хранящие изображение в растровом виде (BMP, TIFF, PCX, PSD, JPEG, PNG, GIF);

хранящие изображение в векторном виде (WMF, AI, CDR);

универсальные, совмещающие векторное и растровое представления (EPS, PICT, FH9, FLA и т. п.).

## 4.4.1 Растровые форматы

*JPEG* — основной формат передачи графики в Интернете. JPEG 2000 способен сжимать файлы практически без потери качества.

PCX — поддерживает RLE-компрессию и цветовую модель RGB, но не поддерживает цветовую модель CMYK.

*PNG* (Portable Network Graphics) — самый перспективный формат Интернета, специально разработанный в качестве замены для GIF. Поддерживает любое количество цветов. Сжатие происходит по принципу, сходному с LZW, но несколько более совершенному. Встроенная гамма-коррекция позволяет правильно отображать цвета на всех компьютерах, независимо от платформы.

RAW — самый гибкий формат для обмена изображениями между компьютерами разных платформ. Может сохранять изображения в любой цветовой модели. Основной формат для цифровых фотоаппаратов.

*TIFF* (Tagged Image File Format) — самый распространенный на сегодняшний день растровый графический формат в издательском деле. Аппаратно независим, поддерживает все цветовые модели, поддерживает LZW-компрессию. Лучший формат для печати.

### 4.4.2 Векторные форматы

DXF (Drawing eXchange Format) предназначен для использования в системах автоматизированного проектирования, прежде всего AutoCAD.

WMF (Windows MetaFile) создавался для обмена данными между Windows-приложениями.

PICT— аналог формата WMF для платформы Macintosh.

## 4.4.3 Универсальные форматы

*CDR* (CorelDRAW) — основной формат программы CorelDRAW. Способен хранить не только элементы векторной графики, но и точечные изображения, текст, эффекты.

Adobe PostScript является стандартом в настольных издательских системах. Использует как векторный, так и растровый способы записи информации.

Разновидности PostScript:

EPS (Encapsulated PostScript).

PDF (Adobe Portable Document Format) — упрощенная и оптимизированная версия PostScript.

## Вопросы для самоконтроля

- 1. Дать понятие информационной технологии.
- 2. Перечислить принципы НИТ.
- 3. Какие операции выполняются при работе с документом?
- 4. Перечислить режимы работы табличного процессора.
- 5. Для чего предназначена СУБД?
- 6. Что представляет реляционная модель данных?
- 7. Перечислить режимы работы PowerPoint.
- 8. Что можно делать с помощью систем компьютерной верстки?
- 9. Перечислить классы СКМ.
- 10. Какие методы применяются для сжатия графической информации?

# **Тема 7 ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ**

#### 1 ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

#### 1.1 Физическая среда передачи информации

Витая пара представляет собой витое двухжильное проводное соединение; легко наращивается. Для повышения помехозащищенности информации используют экранированную витую пару. Скручивание проводов уменьшает влияние внешних электромагнитных колебаний на передаваемые сигналы. Недостаток: плохая помехозащищенность, низкая скорость передачи.

Коаксиальный кабель состоит из внутреннего проводника (стержня) в изоляции и трубки внешнего проводника с защитным покрытием. Хорошо помехозащищен и применяется для связи на расстояние в несколько километров. Для передачи информации на расстояние более 1.5 км требуется репитер. Таким образом, суммарное расстояние при передаче информации может быть увеличено до 10 км.

*Оптоволоконный кабель* состоит из оптического волокна в стеклянном покрытии; сверху защитное покрытие.

#### 1.2 Вычислительные сети

**Сервер** — компьютер, предоставляющий свои услуги пользователям сети.

**Клиент** — компьютер, обращающийся к услугам другой машины.

Сетевая архитектура — представляет собой комбинацию стандартов, топологий и протоколов для создания работоспособной сети.

**Мост** — устройство, соединяющее две сети с одинаковыми методами передачи данных.

**Сетевой адаптер** - устройство, служащее для подключения компьютера к локальной сети.

Компьютерная сеть включает передатчик, сообщение, средства передачи, приемник. Основные компоненты вычислительных сетей приведены на рисунке 1.



Рисунок 1 – Компоненты вычислительных сетей

Для связи отдельных сетей в единую сеть используют мосты. Основные виды компьютеров, подключенных к сети — это сервер и рабочая станция. Сетевой адаптер контролирует доступ к среде передачи данных и обмен данными между единицами сети.

#### 1.3 СЕТЕВЫЕ ПРОТОКОЛЫ

Взаимодействие компьютеров в сети обеспечивается за счет соблюдения *сетевых протоколов* — правил представления и передачи данных, Протоколы семейства TCP/IP представлены на рисунке 2.



обеспечивает негарантированную доставку пакетов без установления соединения между клиентами.

Рисунок 2 – Состав стека ТСР/ІР

Эталонной является модель взаимодействия открытых систем OSI (Open System Interchange). Модель OSI включает 7 уровней протоколов (рисунок 3).

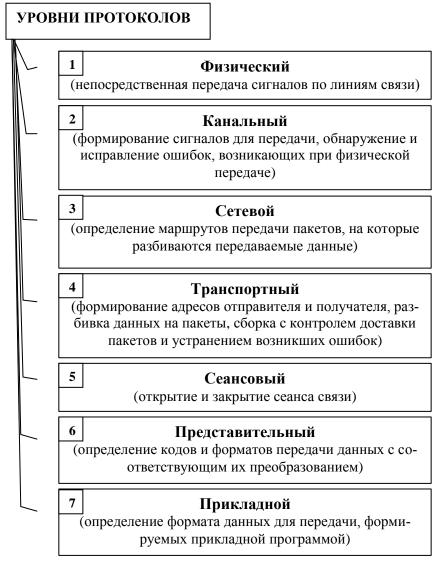


Рисунок 3 – Уровни протоколов в модели OSI

На компьютере отправителя реализуются 7-1 уровни, а на компьютере получателя те же уровни в обратном порядке для восстановления сообщения. На промежуточных компьютерах могут действовать 1-3 уровни.

# 1.4 ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

**Структура (топология) сети** – это способ соединения компьютеров.

Сетевой сегмент — отрезок коаксиального кабеля с подключенными к нему компьютерами; это часть сети или отдельная локальная сеть.

**Концентратор** — устройство, объединяющее несколько каналов связи.

**Ретрансляция** — процесс передачи пакета данных в другую сеть.

**Шлюз** — устройство, обеспечивающее связь между сетями с отличающимися протоколами передачи данных.

**Маршрутиватор** — устройство, обеспечивающее трафик между локальными сетями.

*Покальная вычислительная сеть* (ЛВС) состоит из файлсервера и рабочих станций, подключенных в сетевой сегмент. Сетевая топология определяет технологию передачи данных в сети.

При *шинной* топологии (рисунок 4) среда передачи информации представляет путь, доступный для всех рабочих станций (РС).

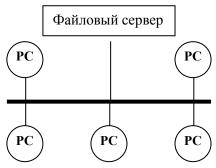


Рисунок 4 – Топология типа «шина»

Рабочие станции в любое время, без прерывания работы всей вычислительной сети, могут быть подключены к ней или отключены. Функционирование вычислительной сети не зависит от состояния отдельной рабочей станции.

Топология типа «звезда» (рисунок 5) предполагает, что головная машина получает и обрабатывает все данные с периферийных устройств. При расширении вычислительной сети к новому рабочему месту необходимо прокладывать отдельный кабель из центра сети.

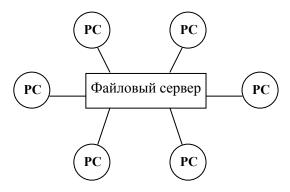


Рисунок 5 – Топология типа «звезда»

Топология в виде звезды является наиболее быстродействующей из всех топологий, поскольку передача данных между рабочими станциями проходит через центральный узел. В случае выхода из строя центрального узла нарушается работа всей сети. Структура типа «шина» проще, экономичнее, но чувствительна к неисправностям кабельной системы. В структуре «звезда» повреждение кабеля на работе сети в целом не сказывается.

При *кольцевой топологии* сети рабочие станции связаны одна с другой по кругу (рисунок 6).

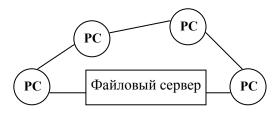


Рисунок 6 – Топология типа «кольцо»

Неисправности в кабельных соединениях локализуются легко.

Ограничение на протяженность вычислительной сети определяется исключительно расстояниями между двумя рабочими станциями.

С помощью мостов из вышеперечисленных типов структур могут быть построены сети со сложной иерархической структурой.

Комбинированная древовидная структура (рисунок7) образуется в виде комбинаций основных топологий вычислительных сетей. Основание дерева вычислительной сети располагается в точке (корень), в которой собираются коммуникационные линии информации (ветви дерева). Вычислительные сети с древовидной структурой применяются там, где невозможно непосредственное применение базовых сетевых структур в чистом виде.

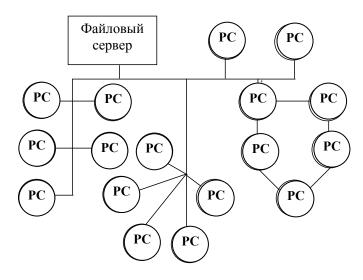


Рисунок 7 – Древовидная структура ЛВС

# 1.5 Глобальные вычислительные сети

**Модем** - устройство, связывающее компьютер с телефонной линией.

Глобальные сети (Wide Area Networks, WAN) предоставляют свои сервисы большому количеству абонентов, разбросанных по большой территории. В затраты на организацию сети входят стоимость кабелей, коммутационного оборудования и усилительной аппаратуры, а также эксплуатационные затраты.

Типичными абонентами глобальной компьютерной сети явля-

ются локальные сети предприятий, расположенные в разных городах и странах, которым нужно обмениваться между собой данными. Основная цель создания сетей заключается в обеспечении обмена данными между двумя вычислительными машинами, входящими в сеть.

Схема связи компьютеров по телефонной линии приведена на рисунке 8.

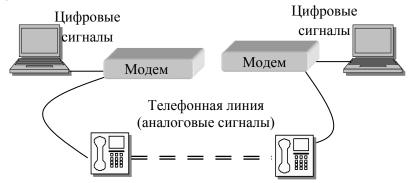


Рисунок 8 – Схема связи компьютеров по телефонной линии

Информация, представленная в компьютере в цифровом виде, с помощью модема преобразуется в аналоговый сигнал, который проходит через аналоговую телефонную линию на телефонную станцию. На телефонной станции аналоговый сигнал преобразуется в цифровую форму и передается по оптоволоконному каналу в уплотненном виде на другую станцию. На другой станции цифровой сигнал вновь преобразуется в аналоговую форму и по аналоговой линии абонента передается к другому модему. Другой модем преобразует аналоговый сигнал в цифровую форму и полученную информацию передает в компьютер. Таким образом, данные на пути к месту назначения проходят два цифро-аналоговых и два аналого-цифровых преобразования.

# 2 ИНТЕРНЕТ

 ${f ISP}$  — Internet Service Provider — поставщик услуг Интернета.

Intranet — это корпоративная сеть, использующая транспортные услуги сети Internet и гипертекстовую технологию.

Internet — это множество компьютеров, находящихся в сети; предоставляет различные способы взаимодействия удаленных компьютеров и совместного использования распределенных услуг и информационных ресурсов.

Host-компьютер - компьютер, подключенный к Internet.

Компьютерная сеть *Интернет* построена на базе компьютеров с операционной системой Unix и протокольного стека TCP/IP. Информация собрана в файлы, которые хранятся на хост-компьютерах, и может быть представлена в различных форматах. Любой компьютер, который поддерживает протоколы TCP/IP, может выступать в качестве хост-компьютера.

Обращаясь в Интернет, мы пользуемся услугами Интернетпровайдера или ISP. Это организация, которая имеет собственную высокоскоростную сеть, объединенную с другими сетями по всему земному шару. ISP-провайдеры — это крупные компании, которые в нескольких населенных пунктах имеют так называемые точки присутствия с аппаратным обеспечением для подключения клиентов к Интернету. Каждый пользователь Интернета имеет договор с определенным провайдером о подключении к его сети. В свою очередь провайдеры договариваются о соединении их сетей. Это позволяет обмениваться сообщениями всем компьютерам, подключенным к Интернету.

#### 2.1 АДРЕСНОЕ ПРОСТРАНСТВО ИНТЕРНЕТА

**Домен** — область адресного пространства Internet.

**Адрес URL** — универсальный указатель на ресурс — является сетевым расширением понятия полного имени ресурса в операционной системе.

Ключом к получению информации в Internet являются адреса ресурсов. Используются почтовые адреса при пересылке сообщений по электронной почте и адреса хост-компьютеров для соединения с ними и для получения файлов с информацией. Для каждого компьютера устанавливается 2 адреса.

1. Цифровой IP-адрес длиной 32 бита, разделенных на 4 блока: 2 блока — адрес сети, 2 блока — адрес компьютера внутри этой сети. Содержит полную информацию необходимую для идентификации компьютера.

**Пример**. 344.40.9.100 (00110101.00101000.00001001.01001100) адрес сети — 344.40, адрес подсети — 9, адрес компьютера — 100

2. Доменный адрес определяет область, представляющую ряд host-компьютеров. В отличие от цифрового читается в обратном порядке. Компьютерное имя включает по крайней мере 2 уровня доменов, которые отделяются друг от друга точкой. Все имена слева от домена верхнего уровня — поддомены общего домена.

**Пример.** Доменный адрес *www.rambler.ru*, здесь *www* — имя сервера, rambler — имя домена второго уровня, ru — имя домена первого уровня.

Домены, представленные географическими регионами, имеют имя, составленное, как правило, из двух букв. Домены, разделенные по тематическим признакам, имеют три буквы в названии.

Пример. companya.msk.ru, window.edu

Россия — ru

edu — образовательное учреждение

gov — правительственное учреждение

сот — коммерческая организация

net — относится к различным компьютерным сетям.

org — используется некоммерческими организациями.

Структура URL приведена на рисунке 1.



Пример. http://www.triton.nsk.ru/struct/news/IVANOV.htm

Первая часть *http://* — это HyperText Transfer Protocol — протокол передачи гипертекста, по которому обеспечивается доставка документа с Web-сервера Web-браузеру.

Вторая часть www.triton.nsk.ru указывает на доменное имя и адресует конкретный компьютер.

Третья часть *struct/news/IVANOV.htm* показывает программеклиенту, где на данном компьютере-сервере искать ресурс: файл *IVANOV.htm* находится в папке *news*, которая расположена в папке *struct*.

#### 2.2 УСЛУГИ ИНТЕРНЕТА

**FTP-сервер** — компьютер, содержащий файлы открытого доступа.

**FTP** (File Transfer Protocol) — протокол передачи файлов.

НТТР - протокол передачи гипертекста.

**НТМL-документы** — текстовые файлы со встроенными командами.

**HTML** - язык гипертекстовой разметки - язык разметки исходного текста веб-документа, включающий специальные символы (теги),

Услуги предоставляемые сетью Internet условно делятся на две категории: обмен информацией между абонентами сети и использование базы данных сети.

FTP-протокол используется для перемещения данных между различными операционными системами, которые могут встретиться в Internet. Докачивание файла из сети Internet после сбоя обеспечивает сервис FTP.

**Электронная почта.** Осуществляет обмен почтовыми сообщениями с любым абонентом сети. Формат адреса:

<uмя пользователя>@<adpec host-компьютера>

Пример. Ivanov@abc.nsk.ru.

Часть слева от значка "@" — это имя почтового ящика (E-mail Account Name) на сервере, из которого владелец адреса забирает письма. Часть справа от значка @ называется доменом и указывает на местонахождение этого почтового ящика.

Как правило, адрес электронной почты определяет не адрес домашнего компьютера пользователя, а адрес сервера, на котором он получает почту.

**Всемирная информационная сеть WWW.** Все серверы WWW используют специальный язык разметки гипертекста HTML.

Связь между гипертекстами осуществляется с помощью ключевых слов. Информация в WWW состоит из страниц (документов). Страницы могут содержать графику, сопровождаться анимацией изображений и звуком, воспроизводимым непосредственно в процессе поступления информации на экран пользователя. Гипертекстовая организация означает, что в документе существуют специальные элементы (текст или рисунки), называемые гипертекстовыми ссылками, щелчок мышью по которым выводит на экран другой документ, на

который указывает данная ссылка. При этом новый документ может храниться на совершенно другом сайте, возможно, расположенном в другом конце земного шара.

Для доступа к web-страницам используется протокол HTTP. Просмотр web-страниц и перемещение между ними в информационном пространстве сети по гиперссылкам обеспечивают специальные программы — браузеры.

# 2.3 Адресация в ІР сетях

Адресация IP обеспечивает пять различных классов сетей (рисунок 2).

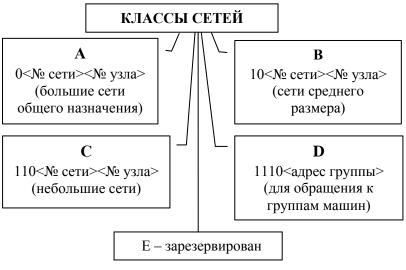


Рисунок 2 – Классы сетей

IP-адресация построена на концепции сети, состоящей из хостов и других сетей. Хост представляет собой объект сети, который может передавать и принимать IP-пакеты. Хосты соединены между собой через одну или несколько сетей. IP-адрес любого хоста состоит из адреса сети и адреса хоста в этой сети. Этот адрес представляется 4 десятичными цифрами, каждое не превышающее 255 (адрес 32 битный). Этот адрес разделяется на 2 или 3 части. Первая часть — адрес сети, вторая (если есть) — адрес подсети, третья — адрес главной вычислительной машины.

Класс	Номер сети, байт	Номер узла, байт	Количество узлов
A	1	3	217-224
В	2	2	28-216
С	3	1	1-28

Первые биты адреса определяют, какая часть адреса относится к номеру сети, а какая к номеру узла.

Некоторые IP адреса являются выделенными и трактуются поособому:

- (все нули) 0.0.0.0 данный узел
- (номер сети).(все нули), например, 194.28.0.0 данная ІР сеть
- (все нули).(номер узла), например, 0.0.0.18 узел в данной локальной сети
- (все единицы) 255.255.255.255 все узлы в данной локальной сети
- (номер сети).(все единицы) например 194.28.255.255 все узлы в указанной IP (локальной) сети
- 127.(что угодно) "петля". Этот адрес используется для тестирования  $\Pi O$  данные не передаются по сети, а возвращаются как только что принятые.

# 2.3.1. Структуризация ІР-сетей с помощью масок

**Маска** — это число, двоичная запись которого содержит единицы в тех разрядах, которые должны интерпретироваться как номер сети.

В случае если количество централизовано выделенных номеров сетей недостаточно для того, чтобы структурировать сеть надлежащим образом, то администратор использует маски, которые позволяют разделять одну сеть на несколько подсетей.

Например, для стандартных классов сетей маски имеют следующие значения:

255.0.0.0 – маска для сети класса А;

255.255.0.0 – маска для сети класса В;

255.255.255.0 – маска для сети класса С.

129.44.0.0 (10000001 00101100 00000000 00000000)

129.44.64.0 (10000001 00101100 01000000 00000000)

129.44.128.0 (10000001 00101100 10000000 00000000)

129.44.192.0 (10000001 00101100 11000000 00000000)

Например, IP-адрес 129.44.141.15 (10000001 00101100 10001101 00001111), который по стандартам IP задает номер сети 129.44.0.0 и номер узла 0.0.141.15, теперь, при использовании маски, будет интерпретироваться как пара:

129.44.128.0 — номер сети, 0.0. 13.15 — номер узла.

# 3 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

«Информационная безопасность — это состояние защищенности информационной среды общества, обеспечивающее ее формирование, использование и развитие в интересах граждан, организаций, государств» (Закон РФ «Об участии в международном информационном обмене»).

Система ващиты информации — организованная совокупность специальных органов, средств, методов и мероприятий, обеспечивающих защиту информации.

**Разглашение** — умышленные или неосторожные действия, приведшие к ознакомлению посторонних лиц с конфиденциальной информацией.

**Утечка** — бесконтрольный выход конфиденциальной информации за пределы организации.

**Несанкционированный доступ** — противоправное овладение конфиденциальной информацией посторонним лицом.

С учетом физической природы путей переноса информации каналы утечки данных можно разбить на группы (рисунок 1).

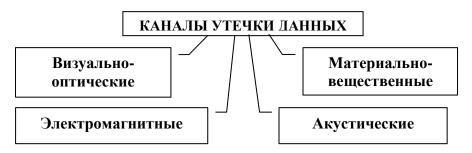


Рисунок 1 – Классификация каналов утечки данных

#### 3.1 ПРАВОВАЯ ЗАЩИТА

Правовая защита информации как ресурса признана на международном, государственном уровне и определяется межгосударственными конвенциями, декларациями. Формы защиты информации приведены на рисунке 2.



Рисунок 2 – Формы правовой защиты информации

Законодательство РФ в области защиты информации:

- 1. Конституция РФ.
- 2. Законы РФ по информатизации и защите информации:
- «Об информации, информатизации и защите информации»;
- «Об участии в международном информационном обмене»;
- «О связи»;
- «Об авторском праве и смежных правах»;
- «О государственной тайне»;
- «О правовой охране программ ЭВМ и топологии микросхем»;
- «Об информационном обеспечении экономического и социального развития»;

- «О страховании»;
- «О коммерческой тайне»;
- «Об органах государственной безопасности»;
- «О патентах»;
- «Об архивах».
- 3. Кодекс об административных правонарушениях, гражданский и уголовный кодексы РФ.
- 4. Нормативно-правовые акты и организационно-распорядительные документы.
- 5. Положения, инструкции, нормативно-технические и методические документы.

#### 3.2 Инженерно-техническая защита

Программные средства защиты — это специальные программы, которые включаются в состав ПО для осуществления функций защиты.

Инженерно-технические средства (ИТС) можно классифицировать по функциональному назначению (рисунок 3).



Рисунок 3 – Классификация ИТС по функциональному назначению

Допуск пользователя и предоставление прав доступа осуществляется через системы идентификации и аутентификации. Идентификация определяет «кто это» и на какие действия имеет право. Аутентификация проверяет подлинность пользователя, действительно ли «он это он».

**Пример.** При входе в систему пользователь вводит свое имя (идентификация) и пароль (аутентификация). В банкоматах: идентификация — ввод карточки, аутентификация — набор PIN (Personal Identification Number) кода.

Криптография включает в себя разделы современной математики, специальные отрасли физики, радиоэлектроники, связи. Криптография должна обеспечивать такую защиту информации, чтобы в случае перехвата ее нельзя было дешифровать в течение нескольких десятков лет.

Защита данных от подделки осуществляется методами шифрования и цифровой подписи на основе криптографических систем с открытым ключом. Электронная цифровая подпись (ЭЦП) устанавливает подлинность информации. Для создания ЭЦП используется сжатый образ исходного текста.

Задача шифрования заключается в добавлении к основному сообщению дополнительных данных так, чтобы выполнялся ряд условий (рисунок 4).



Рисунок 4 – Защита информации шифрованием

Проблемы авторства и отказа от него решаются применением ассиметричного шифрования с открытым ключом. С помощью секретного ключа он шифрует открытое кодовое слово и прилагает к сообщению. Любой получатель может расшифровать эту часть сообщения с помощью открытого ключа. Если при этом действительно получается кодовое слово, значит, шифрование могло быть выполнено только определенным секретным ключом, который может быть известен только предполагаемому автору.

Целостность документа обеспечивается дополнительной шифруемой секретным ключом информацией. В ЭЦП включается дайджест основного документа, например, контрольная сумма или более сложная функция образующих его двоичных цифр. Если после расшифровки она совпадает с реальной характеристикой полученного сообщения, значит, оно не было подменено «по дороге». Таким образом, для формирования цифровой подписи нужна специальная информация отправителя (секретный и открытый ключ, кодовое слово) и сам файл исходного документа (для получения дайджеста). Схема шифрования с ЭЦП приведена на рисунке 5.

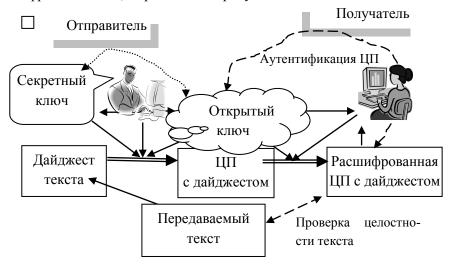


Рисунок 5 – Цифровая подпись

В криптографическом алгоритме с *отврытым ключом* (рисунок 6) имеются как минимум 2 ключа, которые не вычисляются один из другого. Если из ключа шифрования нельзя вычислить ключ расшифрования, то секретность информации, зашифрованной с помощью открытого ключа, обеспечена. Другими словами, если по ключу шифрования не вычисляется ключ расшифрования, то ключ шифрования может быть открытым. Обратное тоже верно.

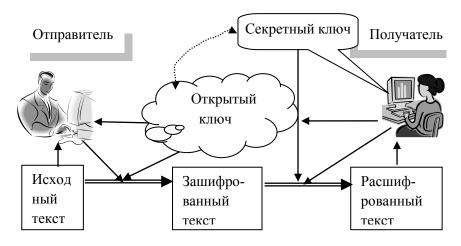


Рисунок 6 – Асимметричное шифрование

Если по ключу расшифрования не вычисляется ключ шифрования, то ключ расшифрования может быть открытым. При этом оба ключа должны быть защищены от подмены и модификации. В двух-ключевой системе шифрования каждый пользователь генерирует свою пару ключей. Текст, зашифрованный одним из них, может быть расшифрован только с использованием второго ключа и наоборот. Закрытый ключ остается у автора, открытый передается абоненту связи.

Схема симметричного шифрования с секретным ключом приведена на рисунке 7.

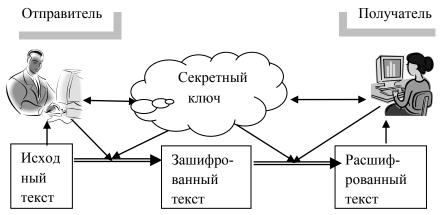


Рисунок 7 – Симметричное шифрование

Перехват данных (подключение к линии связи либо использование побочного электромагнитного излучения средств передачи).

Анализ трафика (наличие/отсутствие связи, частота, направление, объем информации).

*Изменение потока сообщений* (внесение в сообщение не обнаруживаемых искажений, удаление сообщений, нарушение порядка следования).

*Отказ* передающего пользователя от авторства, принимающего — от факта приема сообщений.

Маскарад (стремление выдать себя за другого пользователя с целью получения его прав доступа к информации, либо навязывание другому пользователю ложной информации, якобы исходящей из привычного источника).

Нарушение связи или задержка срочных сообщений.

Классификация угроз безопасности данных приведена на рисунке 8.

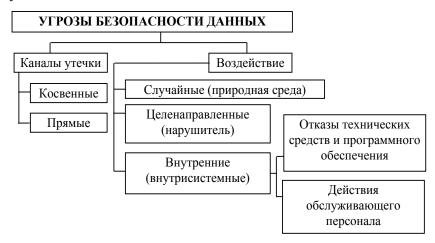


Рисунок 8 – Классификация угроз безопасности данных

#### 3.5 КАТЕГОРИИ КОМПЬЮТЕРНЫХ АТАК

**Компьютерная** атака означает запуск людьми программ для получения неавторизованного доступа к компьютеру.

**Сетевые сканеры** - программы, которые осуществляют сбор информации о сети, чтобы определить, какие из компьютеров и

программ, работающих на них, потенциально уязвимы к атакам.

Сканеры уязвимых мест программ - программы, которые проверяют большие группы компьютеров в Интернете в поисках компьютеров, уязвимых к тому или иному конкретному виду атаки.

**Вскрыватели паролей** — программы, которые обнаруживают легко угадываемые пароли в зашифрованных файлах паролей.

Сетевые анализаторы (снифферы) — программы, которые слушают сетевой трафик. Часто в них имеются возможности автоматического выделения имен пользователей, паролей и номеров кредитных карт из трафика.

*Удаленное проникновение в компьютер*. Программы получают неавторизованный доступ к другому компьютеру через Интернет (или локальную сеть).

*Покальное проникновение в компьютер*. Программы получают неавторизованный доступ к компьютеру, на котором они работают.

Удаленное блокирование компьютера. Программы через Интернет (или локальную сеть) блокируют работу всего удаленного компьютера или отдельной программы на нем.

*Покальное блокирование компьютера*. Программы блокируют работу компьютера, на котором они работают

## 3.6 ВРЕДОНОСНЫЕ ПРОГРАММЫ

Компьютерный вирус — это программа, которая может создавать свои копии (не обязательно совпадающие с оригиналом) и внедрять их в файлы, системные области компьютера, сети и т.д. При этом копии сохраняют способность дальнейшего размножения.

Сетевые черви — это программы, которые не изменяют файлы на дисках, а распространяются в компьютерной сети, проникают в операционную систему компьютера, находят адреса других компьютеров или пользователей и рассылают по этим адресам свои копии.

По способу распространения вредоносные программы (коды) можно условно разделить на 3 класса (рисунок 9).

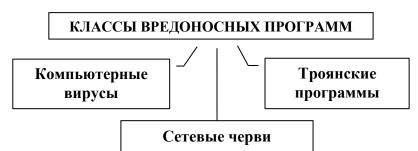


Рисунок 9 — Класси<del>фикация вредоносных програм</del>м по способу распространения

*Компьютерные вирусы* умеют размножаться и внедрять свои копии в другие файлы.

Сетевые черви распространяются по различным сетевым ресурсам (чаще всего по электронной почте), но не внедряют свои копии в другие программы.

*Троянские программы* не распространяются сами по себе, но выполняют на зараженных компьютерах вредоносные действия.

#### 3.6.1 Компьютерные вирусы

В составе программы–вируса выделяют 2 компонента:

- 1. «Голова» начальный код, который перехватывает управление обращение к диску или зараженной вирусом программеносителю.
- 2. «*Тело»* основная часть вируса, осуществляющая копирование и вредоносные действия.

Для того чтобы размножаться, файловый вирус должен копировать свое тело в память, открывать другие исполняемые файлы и записывать туда свое тело.

По характерным особенностям функционирования выделяют 2 класса вирусов (рисунок 10).

Самомаскирующиеся (вирусы-невидимки, стелс (Stealth) вирусы) перехватывают попытки их обнаружить и выдают ложную, маскирующую их присутствие информацию. Например, при запросе длины файла сообщают старую длину до заражения вирусом.

Полиморфные (самомодифицирующиеся, вирусы-мутанты) при копировании в новые заражаемые файлы меняют (шифруют) текст вируса. Это затрудняет его обнаружение по наличию определенных фрагментов кода.



Рисунок 10 — Классификация вирусов по особенностям функционирования

По среде обитания выделяют 4 класса вирусов (рисунок 11).

Сетевые вирусы используют для своего распространения команды и протоколы телекоммуникационных сетей.

Файловые вирусы чаще всего внедряются в исполняемые файлы, имеющие расширение ехе и сот. Также могут внедряться в файлы с компонентами операционных систем, драйверы внешних устройств, объектные файлы и библиотеки, в командные пакетные файлы. При запуске зараженных программ вирус на некоторое время получает управление и производит запланированные действия.



Рисунок 11 – Классификация вирусов по среде обитания

Загрузочные вирусы внедряются в загрузочный сектор дискеты или в главную загрузочную запись жесткого диска. Такой вирус изменяет программу начальной загрузки оперативной системы. В основном это относится к файлам, обеспечивающим доступ пользователей в систему.

Документные вирусы (макровирусы) заражают текстовые файлы редакторов или электронных таблиц, используя макросы, которые сопровождают такие документы. Вирус активизируется, когда документ загружается в соответствующее приложение.

По способу заражения выделяют 2 класса вирусов, приведенные на рисунке 12.



Рисунок 12 – Классификация вирусов по способу заражения

Резидентные вирусы после завершения работы инфицированной программы остаются в оперативной памяти и продолжают свои деструктивные действия. Заражение других исполняемых программ происходит вплоть до выключения компьютера.

*Нерезидентные вирусы* запускаются вместе с зараженной программой и удаляются из памяти вместе с ней.

По деструктивным функциям выделяют 4 класса вирусов (рисунок 13).



Рисунок 13 – Классификация вирусов по деструктивным функциям

По алгоритмам функционирования выделяют 4 класса вирусов (рисунок 14).



Рисунок 14 – Классификация вирусов по алгоритмам функционирования

*Паразитирующие* вирусы изменяют содержимое зараженных файлов. Легко удаляются из файлов.

*Троянские кони* маскируются под полезные программы, нарушают работу системы и собирают сведения, содержащиеся в ней.

*Вирусы-невидимки* способны прятаться при попытке их обнаружения.

*Мутирующие вирусы* периодически изменяют свой программный код, что затрудняет их обнаружение.

# 3.6.2 Сетевые черви

Сетевой червь пытается получить доступ к адресной книге, сканирует жесткий диск на предмет обнаружения любых адресов электронной почты.

По среде распространения выделяют 3 типа (рисунок 15).



Рисунок 15 – Классификация сетевых червей по среде распространения

Бестелесные черви вне компьютера существуют в виде сетевых пакетов, а внутри зараженного компьютера – в виде программного кода прямо в оперативной памяти. Распространяются автоматиче-

ски, используя уязвимости в прикладном и системном программном обеспечении.

# 3.6.3 Троянские программы

Действия злоумышленника на инфицированном ПК:

- кража любой информацию с компьютера-жертвы (файлы, пароли и т.п.);
- проведение любой файловой операции (форматирование жесткого диска, удаление или переименование файлов и т.д.);
- перезагрузка компьютера, подключение к сетевым ресурсам;
- использование зараженного компьютера для атаки на какой-то третий компьютер или сервер в Интернете.

Классификация троянских программ приведена на рисунке 16.



Рисунок 16 – Классификация троянских программ

"Троянцы"-загрузчики переписывают из Интернета и внедряют в систему другие вредоносные коды или вредоносные дополнительные компоненты).

Цель *DoS-атаки*, или атаки типа «отказ в обслуживании», – исчерпать ресурсы информационной системы. В случае успешного проведения DoS-атаки система перестает выполнять свои функции, становится недоступной и иногда непредсказуемой. Чаще всего объектом атаки является web-сервер, например Интернет-магазин.

DDoS-атака, или распределенная атака типа «отказ в обслуживании», отличается от DoS-атаки тем, что в ней один и тот же узел

атакуют сразу несколько компьютеров. Если большое число компьютеров одновременно начнут посылать свои запросы web-серверу, то велика вероятность, что ресурсы web-сервера быстро исчерпаются, а сам сервер не сможет обслуживать легальных пользователей.

Шпионские программы втайне наблюдают за действиями пользователя и записывают в свой журнал интересующие злоумышленника события. Клавиатурные шпионы следят за пользователем и записывают каждое нажатие клавиши. «Троянцы»-шпионы отсылают на удаленный компьютер пароли и другую личную информацию пользователя.

Рекламные программы любым способом пытаются рекламировать продукты или услуги каких-то третьих компаний.

*Программы дозвона* пытаются с помощью модема и телефонной линии дозвониться до платного сервера.

# 3.7 ПРОГРАММНЫЕ СРЕДСТВА ЗАЩИТЫ ОТ КОМПЬЮТЕРНЫХ ВИРУСОВ

Для обнаружения и удаления компьютерных вирусов разработано много различных программ, которые можно разделить на классы (рисунок 17).



Рисунок 17 – Классификация антивирусных программных средств

Детекторы осуществляют поиск компьютерных вирусов в памяти и при обнаружении сообщают об этом пользователю. Выполняют функции мониторинга (отслеживания), проверяют файлы при их загрузке в оперативную память.

Ревизоры запоминают исходное состояние программ, каталогов, системных областей и периодически сравнивают их с текущими значениями. При изменении контролируемых параметров ревизоры сообщают об этом пользователю

Фильтры выявляют подозрительные процедуры, такие как коррекция исполняемых программ, изменение загрузочных записей диска, изменение атрибутов или размеров файла и др. При обнаружении подобных процедур фильтры запрашивают пользователя о правомерности их выполнения.

Доктора являются самым распространенным типом антивирусных программ. Эти программы не только обнаруживают, но и удаляют вирусный код из файла — «лечат». Доктора способны обнаружить и удалить только известные им вирусы, поэтому их необходимо периодически обновлять.

Bакцины модифицируют файл или диск так, что он воспринимается вирусом уже зараженным и поэтому вирус не внедряется.

Особенно важны *полифаги* — программы для поиска большого количества и вирусов различного типа и лечения от них. Они выполняют одновременно функции сканеров и фагов. Использование полифагов требует наличия и постоянного обновления антивирусных баз данных, содержащих признаки максимального количества известных вирусов. Наиболее известные и распространенные полифаги: Антивирус Касперского (Kaspersky AntiVirus, Kaspersky Internet Security), Doctor Web (Dr.Web).

# 3.8 СРЕДСТВА ЗАЩИТЫ СЕТИ ОТ КОМПЬЮТЕРНЫХ АТАК

Межсетевые экраны (брандмауэры или файерволы)— это система, предотвращающая несанкционированный доступ извне во внутреннюю сеть.

Оперативная установка исправлений для программ (Patching). Компании часто выпускают исправления программ, чтобы ликвидировать неблагоприятные последствия ошибок в них.

Обнаружение вирусов и троянских коней. Хорошие антивирусные программы наблюдают за работой компьютеров и выявляют на них вредоносные программы. Для максимальной эффективности они должны быть установлены на всех компьютерах в сети.

Межсетевые экраны — самое важное средство защиты сети — контролируют сетевой трафик, входящий в сеть и выходящий из нее, могут блокировать передачу в сеть какого-либо вида трафика. Хорошо сконфигурированный межсетевой экран в состоянии остановить большинство известных компьютерных атак.

Аппаратный брандмауэр подключается к сети физически, фильтрует входящий и исходящий трафик и защищает от нежелательных проникновений во внутреннюю сеть или на персональный ком-

пьютер. Программный брандмауэр выполняет те же функции, но является не внешним аппаратным устройством, а программой. Вскрыватели паролей. Это средство будет полезно системным

*Вскрыватели паролей*. Это средство будет полезно системным администраторам, для своевременного обнаружения слабых паролей.

*Шифрование*. Соединения с удаленными машинами, защищаемые с помощью пароля, должны быть зашифрованы.

Сканеры уязвимых мест. Это программы, которые сканируют сеть в поисках компьютеров, уязвимых к определенным видам атак. Грамотное конфигурирование компьютеров в отношении

Грамотное конфигурирование компьютеров в отношении безопасности. Компьютеры с заново установленными операционными системами часто уязвимы к атакам, так как при начальной установке операционной системы обычно разрешаются все сетевые средства.

Боевые диалеры(war dialer). Пользователи часто обходят средства защиты сети организации, разрешая своим компьютерам принимать входящие телефонные звонки. Боевые диалеры используются для проверки телефонных номеров пользователей и обнаружения сконфигурированных подобным образом компьютеров.

Рекомендации по безопасности — это предупреждения, публикуемые группами по борьбе с компьютерными преступлениями и производителями программ о недавно обнаруженных уязвимых местах. Они описывают в целом угрозу и дают довольно конкретные советы о том, что нужно сделать для устранения уязвимого места.

Средства обнаружения атак (Intrusion Detection) могут быть установлены за межсетевым экраном, чтобы обнаруживать атаки, организуемые изнутри сети. Также они могут быть установлены перед межсетевым экраном, чтобы обнаруживать атаки на сам экран.

Средства выявления топологии сети и сканеры портов. Эти программы позволяют составить полную картину того, как устроена сеть и какие компьютеры в ней работают, а также выявить все сервисы, которые работают на каждой машине. Системные администраторы используют эти средства для наблюдения за тем, какие программы и на каких компьютерах работают в их сети.

Политики безопасности. Если в рамках одной организации имеется несколько сетей с различными политиками безопасности, то одна сеть может быть скомпрометирована из-за плохой безопасности другой сети. Компоненты политики безопасности приведены на рисунке 18.

# Пропускаемый трафик Средства защиты Стандартные безопасные конфигурации для различных типов компьютеров

Рисунок 18 – Компоненты политики безопасности

#### Вопросы для самоконтроля

- 1. Какие существуют среды передачи информации?
- 2. Для чего нужен протокол?
- 3. Какие существуют типы топологий в сетях?
- 4. Какие функции выполняет модем?
- 5. Чем цифровой IP-адрес отличается от доменного?
- 6. Какие услуги предоставляет Интернет?
- 7. Какие существуют формы правовой защиты информации?
- 8. Для чего нужна электронная цифровая подпись?
- 9. Какие программные средства используются для защиты от компьютерных вирусов?
- 10. Перечислить средства защиты сети от компьютерных атак?

- 1. Андреева Е., Фалина И. Системы счисления и компьютерная арифметика.—М.: Лаборатория Базовых Знаний, 1999. 256 с.
- 2. Акулов О., Медведев Н. Информатика: базовый курс: учебное пособие для студентов вузов. 2–е изд., испр. и доп. М.: Омега–Л, 2006, 552 с.
- 3. Гук М.Ю. Интерфейсы устройств хранения. Энциклопедия.— СПб.: Питер, 2007.— 448 с.
- 4. Информатика: Учебник/Под ред. Н. В. Макаровой.–М.: Финансы и статистика, 2009.–768 с.
- 5. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учеб.для вузов. 4-е изд. СПб.: Питер, 2010. 943 с.
- 6. Олифер В. Г., Олифер Н. А. Сетевые операционные системы: Учеб.для вузов. 2–е изд.— СПб.: Питер, 2009.— 669 с.
- 7. Острейковский. Информатика: Учеб. для техн. направлений и спец. вузов.— М.: Высш. шк., 2009.— 512 с.
- 8. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 816 с.
- 9. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд.— СПб.: Питер, 2012.— 960 с.
- 10. Таненбаум Э. Современные операционные системы. 4-е изд.— СПб.: Питер, 2015.— 1120 с.
- 11. Шаньгин В. Ф. Информационная безопасность. ДМК Пресс, 2014. 702 с.
- 12. Ярочкин В. И. Информационная безопасность.— М.: Академический проект, Гаудемус, 2008.— 544 с.
- 13. Материалы сайтов www.intuit.ru, https://overclockers.ru, https://3dnews.ru.

ПРЕДИСЛОВИЕ	3
Тема 1 ВВЕДЕНИЕ В ИНФОРМАТИКУ	
1 ВВЕДЕНИЕ	
1.1 Информатика, как естественнонаучная дисциплина	
*1.2 История развития вычислительной техники	
*1.3 Общая характеристика информационных процессов	6
*1.4 Информационный рынок	
1.5 Информация: понятие, свойства	9
1.6 Кодирование текстовой информации	
2 ИЗМЕРЕНИЕ КОЛИЧЕСТВА ИНФОРМАЦИИ	
2.1 Меры и единицы измерения информации	12
*2.2 Положения комбинаторики, используемые в измерении	
информации	15
Вопросы для самоконтроля Тема 2 ЛОГИЧЕСКИЕ И АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЭВМ	17
1 ЛОГИЧЕСКИЕ ОСНОВЫ ЭВМ	
1.1 Основные понятия алгебры логики	
1.2 Формализация высказываний	
1.3 Логические схемы	20
2 АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЭВМ	
2.1 Системы счисления	
<ol> <li>2.2 Арифметические действия в позиционных системах счислени</li> <li>22</li> </ol>	
2.3 Сложение по модулю	24
3 ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ЭВМ	
3.1 Формы представления чисел	25
*3.2 Особенности представления чисел в ЭВМ	27
3.3 Прямой, обратный, дополнительный коды	27
3.4 Модифицированные коды	
Вопросы для самоконтроля	33
Тема 3 ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ	
СИСТЕМ	
1 УСТРОЙСТВО ЭВМ	
1.1 Состав и назначение основных блоков	
*1.2 Функционирование вычислительных машин	
1.3 Процессоры	
*1.4 Типы и характеристики интерфейсов	
1.5 Основные устройства	44
*1.6 Периферийное оборудование	49
2 УСТРОЙСТВА ХРАНЕНИЯ ИНФОРМАЦИИ	51
2.1 Классификация устройств	
2.2 Характеристика устройств памяти	51
*2.3 Носители информации	

вопросы для самоконтроляТема 4 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕ	58
Тема 4 ТРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕ СИСТЕМ	
1 АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ	59
1.1 Понятие алгоритма	59
*1.2 Запись алгоритма	
*1.3 Структуры алгоритмов	
*1.4 Структуры данных	
2 СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	66
2.1 Классификация программных продуктов	
2.2 Файловая система диска	
*2.3 Система прерываний	70
3 ОПЕРАЦИОННЫЕ СИСТЕМЫ СЕМЕЙСТВА WINDOWS	
3.1 История создания графической операционной среды	ı71
3.2 Графический пользовательский интерфейс	
3.3 Оконный интерфейс Windows	
3.4 Объекты пользовательского уровня	
3.5 Обмен данными в среде Windows	76
*4 ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ	
4.1 Основные понятия	
4.2 Компиляция, отладка и выполнение программ	
4.3 Объектно-ориентированное программирование	
4.4 Обзор языков программирования	
Вопросы для самоконтроля	86 -
Тема 5 ПРЕДСТАВЛЕНИЕ, КОДИРОВАНИЕ, КОНТРОЛ	ь
ПЕРЕДАЧИ ИНФОРМАЦИИ 1 СПОСОБЫ ПРЕДСТАВЛЕНИЯ И МЕТОДЫ	
НФОРМАЦИИ ПО КАНАЛАМ СВЯЗИ	негедачи 87
1.1 Представление информации физическими сигналам	
1.2 Методы передачи информации	87
*1.3 Характеристика каналов связи	
*1.4 Передача информации по каналам связи	
*1.5 Помехи в каналах связи	
*1.6 Коммутация в сетях	
2 КОДИРОВАНИЕ И КОНТРОЛЬ ПЕРЕДАЧИ ИНФОРМАЦ	<b>ДИИ 97</b>
2.1 Основные понятия	
*2.2 Классификация двоичных кодов	
2.3 Характеристики кодов	
*2.4 Методы кодирования информации	
2.5 Контроль передачи информации	102
2.6 Помехоустойчивое кодирование	
3 СЖАТИЕ ИНФОРМАЦИИ	
*3.1 Простейшие алгоритмы сжатия информации	
*3.2 Особенности программ-архиваторов	111

*3.3 Сжатие информации с потерями	.112
Вопросы для самоконтроля	.113
Тема 6 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЕ	
ТЕХНОЛОГИЙ 1 КОМПЬЮТЕРНЫЕ ОФИСНЫЕ ТЕХНОЛОГИИ	.114
1 КОМПЬЮТЕРНЫЕ ОФИСНЫЕ ТЕХНОЛОГИИ	115
1.1 Текстовый процессор	.115
1.2 Электронная таблица	.118
1.3 Система управления базами данных	.124
1.4 Редактор презентаций	.132
2 СИСТЕМЫ КОМПЬЮТЕРНОЙ ВЕРСТКИ	134
2.1 Система верстки ТеХ	
2.2 Система верстки Scribus	
3 СИСТЕМЫ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ	
4 КОМПЬЮТЕРНАЯ ГРАФИКА	
4.1 Разновидности компьютерной графики	
4.2 Цветовые модели	.139
4.3 Сжатие графических изображений	.141
4.4 Графические форматы	.142
Вопросы для самоконтроля	.143
Тема 7 ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ	
1 ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ	
1.1 Физическая среда передачи информации	
1.2 Вычислительные сети	
1.3 Сетевые протоколы	
1.4 Локальные вычислительные сети	
1.5 Глобальные вычислительные сети	
2 ИНТЕРНЕТ	
2.1 Адресное пространство Интернета	
2.2 Услуги Интернета	
2.3 Адресация в IP сетях	
3 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ	
3.1 Правовая защита	
3.2 Инженерно-техническая защита	
3.3 Криптографические методы защиты информации	
3.4 Угрозы безопасности информации при передаче	
3.5 Категории компьютерных атак	
3.6 Вредоносные программы	
3.7 Программные средства защиты от компьютерных вирусов	
3.8 Средства защиты сети от компьютерных атак	
Вопросы для самоконтроля	.172