

Министерство сельского хозяйства Российской Федерации  
ФГБОУ ВО «Красноярский государственный аграрный университет»

*Н.В. Титовская, С.Н. Титовский,  
И.И. Болдарук, Н.Д. Амбросенко*

## **ВВЕДЕНИЕ В ИНФОРМАТИКУ (БАЗОВЫЙ КУРС)**

*Рекомендовано учебно-методическим советом федерального государственного бюджетного образовательного учреждения высшего образования «Красноярский государственный аграрный университет» для внутривузовского использования в качестве учебного пособия для студентов, обучающихся по всем специальностям, направлениям и формам подготовки, а также для слушателей курсов при подготовке к вступительным экзаменам по информатике*

*Электронное издание*

Красноярск 2022

ББК 32.973я73

Т 45

*Рецензенты:*

*С.В. Ченцов, д-р техн. наук, проф. каф. СААУП ИКИТ СФУ*

*И.А. Панфилов, канд. техн. наук, доцент каф. САИО  
СибГТУ им. М.Ф. Решетнева*

Т 45 **Титовская, Н.В.**  
**Введение в информатику (базовый курс)** [Электронный ресурс]: учебное пособие / Н.В. Титовская, С.Н. Титовский, И.И. Болдарук, Н.Д. Амбросенко; Красноярский государственный аграрный университет. – Красноярск, 2022. – 315 с.

Пособие содержит теоретические положения и практические задания по информатике, а также контрольные вопросы и тестовые задания по пройденному материалу.

Предназначено для самостоятельного освоения базового курса дисциплины «Информатика» студентами всех форм обучения, направлений подготовки и специальностей, а также для слушателей курсов при подготовке к вступительным экзаменам по информатике.

ББК 32.973я73

© Титовская Н.В., Титовский С.Н.,  
Болдарук И.И., Амбросенко Н.Д., 2022  
© ФГБОУ ВО «Красноярский государственный  
аграрный университет», 2022

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
Глава 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ .....	8
1.1. ПОНЯТИЕ ТЕРМИНА «ИНФОРМАТИКА».....	8
1.2. ПОНЯТИЕ ТЕРМИНА «ИНФОРМАЦИЯ». ИЗМЕРЕНИЕ КОЛИЧЕСТВА ИНФОРМАЦИИ .....	9
1.3. КАК ПЕРЕДАЕТСЯ И ОБРАБАТЫВАЕТСЯ ИНФОРМАЦИЯ.....	12
1.4. СВОЙСТВА ИНФОРМАЦИИ И ОСНОВНЫЕ ОПЕРАЦИИ, ВЫПОЛНЯЕМЫЕ С ИНФОРМАЦИЕЙ .....	12
1.5. ИНФОРМАЦИОННЫЕ РЕСУРСЫ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ...	14
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	14
Глава 2. КОДИРОВАНИЕ ЧИСЛОВОЙ И СИМВОЛЬНОЙ ИНФОРМАЦИИ, КОДОВЫЕ ТАБЛИЦЫ .....	16
2.1. КОДИРОВАНИЕ ДАННЫХ ДВОИЧНЫМ КОДОМ.....	16
2.1.1. Кодирование целых и действительных чисел, текстовой информации.....	16
2.1.2. Кодирование графических данных .....	18
2.1.3. Кодирование звуковой информации .....	21
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	22
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	23
Глава 3. ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ И РАБОТЫ КОМПЬЮТЕРОВ.....	28
3.1. ОСНОВНЫЕ ПОНЯТИЯ О КОМПЬЮТЕРЕ .....	28
3.2. УСТРОЙСТВО КОМПЬЮТЕРА.....	28
3.3. ПРИНЦИПЫ ПОСТРОЕНИЯ КОМПЬЮТЕРА .....	30
3.4. КОМАНДА КОМПЬЮТЕРА .....	32
3.5. АРХИТЕКТУРА И СТРУКТУРА КОМПЬЮТЕРА .....	33
3.6. УСТРОЙСТВО ПАМЯТИ КОМПЬЮТЕРА.....	36
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	37
Глава 4. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА.....	39
4.1. УСТРОЙСТВА, ВХОДЯЩИЕ В СОСТАВ СИСТЕМНОГО БЛОКА.....	41
4.1.1. Материнская плата.....	41
4.1.2. Центральный процессор.....	41
4.1.3. Устройства, образующие внутреннюю память.....	47
4.1.4. Жесткий диск, или винчестер .....	51
4.1.5. Графическая плата .....	53
4.1.6. Звуковая плата .....	54
4.1.7. Сетевая плата .....	55

4.2. ПЕРИФЕРИЙНЫЕ ВНЕШНИЕ УСТРОЙСТВА.....	58
4.3. КОНФИГУРАЦИЯ КОМПЬЮТЕРА .....	78
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	78
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	81
Глава 5. ОРГАНИЗАЦИЯ МЕЖКОМПЬЮТЕРНОЙ СВЯЗИ .....	84
5.1. СПОСОБЫ ОРГАНИЗАЦИИ МЕЖКОМПЬЮТЕРНОЙ СВЯЗИ .....	84
5.2. ПОНЯТИЕ О КОМПЬЮТЕРНОЙ СЕТИ.....	85
5.3. СОЕДИНЕНИЕ УСТРОЙСТВ СЕТИ.....	87
5.4. СЕТЬ ИНТЕРНЕТ .....	89
5.4.1. Подключения к сети Интернет .....	92
5.4.2. Пересылка данных в Интернет. Протоколы связи TCP/IP .....	93
5.4.3. Адресация в Интернете .....	94
5.4.4. Обзор сервисов Интернета .....	98
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	113
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	115
Глава 6. ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ. КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ.....	117
6.1. ИСТОРИЯ РАЗВИТИЯ СРЕДСТВ ОБРАБОТКИ ИНФОРМАЦИИ .....	117
6.2. ЭТАПЫ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ. ПОКОЛЕНИЯ ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН .....	137
6.3. ТИПЫ И НАЗНАЧЕНИЕ КОМПЬЮТЕРОВ.....	149
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	150
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	151
Глава 7. СИСТЕМЫ СЧИСЛЕНИЯ. АРИФМЕТИКА В РАЗЛИЧНЫХ СИСТЕМАХ СЧИСЛЕНИЯ.....	152
7.1. СИСТЕМЫ СЧИСЛЕНИЯ .....	152
7.2. ПЕРЕВОД ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ.....	157
7.3. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ, ВЫПОЛНЯЕМЫЕ В ПОЗИЦИОННЫХ СИСТЕМАХ СЧИСЛЕНИЯ.....	162
7.4. КОДИРОВАНИЕ ИНФОРМАЦИИ .....	164
7.4.1. Представление чисел в компьютере.....	166
7.4.2. Сложение и вычитание двоичных чисел .....	168
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	170
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	172
Глава 8. АЛГЕБРА ЛОГИКИ.....	178
8.1. ВОЗНИКНОВЕНИЕ ЛОГИКИ КАК САМОСТОЯТЕЛЬНОЙ НАУКИ .....	178
8.2. АЛГЕБРА ЛОГИКИ КАК НАУКА ОБ ОБЩИХ ОПЕРАЦИЯХ НАД ЛОГИЧЕСКИМИ ВЫСКАЗЫВАНИЯМИ .....	181
8.3. ПОНЯТИЕ ЛОГИЧЕСКОЙ ФОРМУЛЫ. ЗАКОНЫ АЛГЕБРЫ ЛОГИКИ ....	185

8.4. ТАБЛИЦЫ ИСТИННОСТИ .....	187
8.5. СИСТЕМЫ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ .....	189
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	191
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	192
Глава 9. АЛГОРИТМЫ. АЛГОРИТМИЗАЦИЯ.	
АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ .....	198
9.1. АЛГОРИТМ. СВОЙСТВА АЛГОРИТМОВ .....	198
9.2. ФОРМЫ ЗАПИСИ АЛГОРИТМОВ.....	200
9.3. ГРАФИЧЕСКАЯ ФОРМА ЗАПИСИ АЛГОРИТМА.....	202
9.3.1. <i>Виды алгоритмов</i> .....	205
9.4. ЯЗЫКИ ПРОГРАММИРОВАНИЯ .....	214
9.4.1. <i>Программный способ записи алгоритмов. Уровни языка программирования</i> .....	214
9.4.2. <i>Процедурно-ориентированное программирование</i> .....	215
9.4.3. <i>Объектно-ориентированное программирование</i> .....	217
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ.....	226
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	228
Глава 10. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА .....	
10.1. КЛАССИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	239
10.2. ОПЕРАЦИОННЫЕ СИСТЕМЫ.....	241
10.3. ФАЙЛЫ И ФАЙЛОВАЯ СИСТЕМА.....	247
10.4. СЕРВИСНОЕ ПО (СЛУЖЕБНЫЕ ПРОГРАММЫ).....	250
10.5. КОМПЬЮТЕРНЫЕ ВИРУСЫ И АНТИВИРУСНЫЕ СРЕДСТВА.....	252
10.5.1. <i>Компьютерные вирусы</i> .....	252
10.5.2. <i>Антивирусные мероприятия</i> .....	258
10.6. ИНСТРУМЕНТАЛЬНЫЕ СИСТЕМЫ ПРОГРАММИРОВАНИЯ .....	260
10.7. ДРАЙВЕРЫ .....	261
10.8. АРХИВАТОРЫ .....	262
10.9. ПРОГРАММЫ ОБСЛУЖИВАНИЯ ЖЕСТКИХ ДИСКОВ .....	265
10.10. ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ .....	266
10.10.1. <i>Средства обработки текстовой информации</i> .....	266
10.10.2. <i>Средства обработки табличной информации</i> .....	268
10.10.3. <i>Средства обработки графической информации</i> .....	270
10.10.4. <i>3D-графика</i> .....	276
10.10.5. <i>Системы управления базами данных (СУБД)</i> .....	277
10.10.6. <i>Средства разработки презентаций</i> .....	278
10.10.7. <i>Автоматизация ввода информации в компьютер</i> .....	279
10.10.8. <i>Автоматизация перевода текста</i> .....	279
10.10.9. <i>Издательские системы</i> .....	280

<i>10.10.10. Системы автоматизации бухгалтерской деятельности</i> .....	281
<i>10.10.11. Прочее программное обеспечение</i> .....	281
ВОПРОСЫ ДЛЯ ПОВТОРЕНИЯ И САМОКОНТРОЛЯ .....	281
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	284
СЛОВАРЬ ОСНОВНЫХ ПОНЯТИЙ И ТЕРМИНОВ .....	287
ЗАКЛЮЧЕНИЕ .....	311
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	312

## ВВЕДЕНИЕ

Информатика – наука о методах и процессах сбора, хранения, обработки, передачи, анализа и оценки информации с применением компьютерных технологий, обеспечивающих возможность ее использования для принятия решений.

Информатика нацелена на разработку общих методологических принципов построения информационных моделей. Поэтому методы информатики применимы всюду, где существует возможность описания объекта, явления, процесса и т. п. с помощью информационных моделей.

В учебном пособии изложены информационные основы, технические и программные средства реализации информационных процессов, компьютерные сети, основы алгоритмизации и программирования. Достаточно подробно рассматриваются приложения *Microsoft Office*: текстовый процессор *Word*, табличный процессор *Excel*, система управления базами данных *Access*, программа подготовки презентаций *PowerPoint*.

Особое внимание уделяется вопросам алгоритмизации и программирования. Освещение этих вопросов сопровождается большим количеством примеров, наглядно показывающих различные способы программирования вычислительных процессов и характерные особенности в применении тех или иных алгоритмов.

Структура учебного пособия позволяет заострить внимание студентов на проблемных и перспективных вопросах, последовательно освоить учебный материал. Представленные в пособии задания для самостоятельной работы студентов позволяют закрепить теоретические знания и сформировать практические навыки, большое количество визуального материала, а также способствует лучшему усвоению материала.

# Глава 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

## 1.1. Понятие термина «информатика»

Термин «информатика» (франц. *informatique*) происходит от французских слов *information* (информация) и *automatique* (автоматика) и дословно означает «информационная автоматика».

Широко распространен также англоязычный вариант этого термина – «*Computer science*», что означает буквально «компьютерная наука».

Информатика – это основанная на использовании компьютерной техники дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы ее создания, хранения, поиска, преобразования, передачи и применения в различных сферах человеческой деятельности [13].

Информатика базируется на компьютерной технике и немыслима без нее.

Российский академик А.А. Дородницын выделяет в информатике три неразрывно и существенно связанные части: технические средства, программные и алгоритмические [13].

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

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

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



## 1.2. Понятие термина «информация». Измерение количества информации

Термин «информация» происходит от латинского слова «*informatio*», что означает сведения, разъяснения, изложение. Несмотря на широкое распространение этого термина, понятие информации является одним из самых дискуссионных в науке [13].

Можно выделить, по крайней мере, четыре различных подхода к определению понятия «информация».

В первом слово *информация* применяется как синоним интуитивно понимаемых слов: *сведения, знания, сообщение, осведомление о положении дел*.

Во втором, кибернетическом, понятие информация используется для характеристики управляющего сигнала, передаваемого по линии связи.

В третьем, философском, понятие *информация* тесно связано с такими понятиями, как взаимодействие, отражение, познание.

Наконец, в четвертом, вероятностном, информация вводится как мера уменьшения неопределенности и позволяет количественно измерять информацию, что чрезвычайно важно для информатики как технологической науки.

Клод Шеннон, американский ученый, заложивший основы теории информации – науки, изучающей процессы, связанные с передачей, приемом, преобразованием и хранением информации, – рассматривает информацию как снятую неопределенность наших знаний о чем-то.

Количество информации в этой теории определяется по формуле

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

где  $I$  – количество информации,

$n$  – количество возможных событий,

$p_i$  – вероятности отдельных событий.

Пусть потенциально может осуществиться некоторое множество событий ( $n$ ), каждое из которых может произойти с некоторой вероятностью ( $p_i$ ), т. е. существует неопределенность. Предположим, что одно из событий произошло, неопределенность уменьшилась, вернее, наступила полная определенность. Количество информации ( $I$ ) является мерой уменьшения неопределенности.

Для частного, но широко распространенного случая, когда события равновероятны ( $p_i = 1/n$ ), величина количества информации  $I$  принимает максимальное значение:

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

Для измерения количества информации нужна единица измерения. За единицу количества информации приняли такое количество информации, при котором неопределенность уменьшается в два раза, т. е., например, когда в простейшем случае из двух возможных событий реализуется одно:

$$I = \log_2 2 = 1 \text{ бит.}$$

Эта единица измерения информации получила название *бит* (*bit* – от английского словосочетания *Binary digit*).

Например, при бросании монеты существует два равновероятных исхода (события): орел или решка. Монета упала, событие произошло, количество информации равно 1 бит. Таким ситуациям с двумя возможностями приписывается начальная неопределенность:  $A = 1$ .

После бросания монеты конечная неопределенность  $K = 0$ .

Количество информации, полученное при бросании:

$$I = A - K = 1 - 0 = 1.$$

В детской игре «Угадай число» первый игрок загадывает число (например, в диапазоне от 1 до 100), второй задает вопросы типа: «Число больше 50?» Ответ (да или нет) несет информацию 1 бит, так как неопределенность (количество возможных событий) уменьшается в два раза. Оптимальная стратегия отгадывания состоит в делении на каждом шаге массива возможных чисел пополам. Действительно, именно в случае равновероятных событий (одинаковых по объему массивов чисел) количество информации имеет максимальное значение.

При бросании шестигранного кубика может произойти шесть событий, а при бросании шарика в рулетке – тридцать шесть. Количество возможных событий  $N$  и количество информации  $I$  связаны между собой следующей зависимостью:  $N = 2^I$ .

Так, при бросании равносторонней четырехгранной пирамиды (тетраэдра) вероятность ожидаемых событий равна четверем.  $4 = 2^2$ .

Тогда:  $I = 2$ . Таким образом, при бросании пирамидки получена информация, равная двум битам.

В вычислительной технике битом называют наименьшую «порцию» памяти, необходимую для хранения одного из двух знаков «0» и «1», используемых для внутримашинного представления данных и команд.

Бит – слишком мелкая единица измерения. На практике чаще применяется более крупная единица – байт, равная восьми битам. Именно восемь битов требуется для того, чтобы закодировать любой из 256 символов алфавита клавиатуры компьютера ( $256=2^8$ ).

Широко используются также более крупные производные единицы информации:

1 Килобайт (Кбайт) = 1024 байт =  $2^{10}$  байт;

1 Мегабайт (Мбайт) = 1024 Кбайт =  $2^{20}$  байт;

1 Гигабайт (Гбайт) = 1024 Мбайт =  $2^{30}$  байт.

В последнее время в связи с увеличением объемов обрабатываемой информации входят в употребление такие производные единицы:

1 Терабайт (Тбайт) = 1024 Гбайт =  $2^{40}$  байт;

1 Петабайт (Пбайт) = 1024 Тбайт =  $2^{50}$  байт;

1 Экзабайт (Эбайт) = 1024 Пбайта =  $2^{60}$  байт;

1 Зеттабайт (Збайт) = 1024 Эбайта =  $2^{70}$  байт;

1 Йоттабайт (Йбайт) = 1024 Збайта =  $2^{80}$  байт.

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

*Решение:*

1 страница книги содержит  $35 \times 50 = 1750$  символов или 1750 байт информации (1 символ = 1 байт).

Объем всей информации в книге (в разных единицах):

$1750 \times 100 = 175000$  байт.

$175000 / 1024 = 170,8984$  Кбайт.

$170,8984 / 1024 = 0,166893$  Мбайт.

За единицу информации можно было бы выбрать количество информации, необходимое для различения, например, десяти равновероятных сообщений. Это будет не двоичная (бит), а десятичная (дит) единица информации.

### 1.3. Как передается и обрабатывается информация

Информация передается в виде сообщений от некоторого источника информации (управляемого объекта) к ее приемнику (управляющему органу) посредством канала связи между ними. Источник посылает передаваемое сообщение, которое кодируется в передаваемый сигнал. Этот сигнал посылается по каналу прямой связи. В результате в приемнике появляется принимаемый сигнал, который декодируется и становится принимаемым сообщением (рис. 1.1).

Если источник и приемник соединены каналами прямой и обратной связи, то такую систему называют замкнутой или системой с обратной связью.



Рисунок 1.1 – Система с обратной связью

По каналу прямой связи передаются сигналы (команды) управления, вырабатываемые в управляющем органе. Подчиняясь этим командам, управляемый объект осуществляет свои рабочие функции. В свою очередь, управляемый объект соединен с управляющим органом каналом обратной связи, по которому поступает информация о состоянии управляемого объекта. В управляющем органе эта информация используется для выработки новых сигналов управления, направляемых к управляемому объекту.

### 1.4. Свойства информации и основные операции, выполняемые с информацией

Свойства информации:

- достоверность;
- полнота;
- ценность;
- своевременность;
- понятность;
- доступность;
- краткость;
- и др.

Информация достоверна, если она отражает истинное положение дел. Недостоверная информация может привести к неправильному пониманию или принятию неправильных решений.

Достоверная информация со временем может стать недостоверной, так как она обладает свойством устаревать, т. е. перестает отражать истинное положение дел.

Информация полна, если ее достаточно для понимания и принятия решений. Как неполная, так и избыточная информация сдерживает принятие решений, или может повлечь ошибки.

Точность информации определяется степенью ее близости к реальному состоянию объекта, процесса, явления и т. п.

Ценность информации зависит от того, насколько она важна для решения задачи, а также от того, насколько в дальнейшем она найдет применение в каких-либо видах деятельности человека.

Только своевременно полученная информация может принести ожидаемую пользу. Одинаково нежелательны как преждевременная подача информации (когда она еще не может быть усвоена), так и ее задержка.

Если ценная и своевременная информация выражена непонятным образом, она может стать бесполезной.

Информация становится понятной, если она выражена языком, на котором говорят те, кому предназначена эта информация.

Информация должна преподноситься в доступной (по уровню восприятия) форме. Поэтому одни и те же вопросы по-разному изложены в школьных учебниках и научных изданиях.

Информацию по одному и тому же вопросу можно изложить кратко (сжато, без несущественных деталей) или пространно (подробно, многословно). Краткость информации необходима в справочниках, энциклопедиях, учебниках, всевозможных инструкциях.

#### *Основные операции с информацией*

Информацию можно:

- создавать;
- передавать;
- воспринимать;
- использовать;
- запоминать;
- принимать;
- копировать;
- формализовать;
- распространять;
- преобразовывать;
- комбинировать;
- обрабатывать;
- делить на части;
- упрощать;
- собирать;
- хранить;
- искать;
- измерять;
- разрушать;
- и др.

Все эти процессы, связанные с определенными операциями над информацией, называются информационными процессами.

## **1.5. Информационные ресурсы и информационные технологии**

Информационные ресурсы – это идеи человечества и указания по их реализации, накопленные в форме, позволяющей их воспроизводство.

Это книги, статьи, патенты, диссертации, научно-исследовательская и опытно-конструкторская документация, технические переводы, данные о передовом производственном опыте и др.

Информационные ресурсы (в отличие от всех других видов ресурсов – трудовых, энергетических, минеральных и т. д.) тем быстрее растут, чем больше их расходуют.

Информационная технология – это совокупность методов и устройств, используемых людьми для обработки информации.

Человечество занималось обработкой информации тысячи лет. Первые информационные технологии основывались на использовании счетов и письменности. Быстрое развитие этих технологий связано с появлением компьютеров.

В настоящее время термин «информационная технология» употребляется в связи с использованием компьютеров для обработки информации. Информационные технологии охватывают всю вычислительную технику и технику связи, бытовую электронику, телевидение и радиовещание.

### **Вопросы для повторения и самоконтроля**

1. Что означает термин «информатика»? Каково его происхождение?
2. Какие области знаний и административно-хозяйственной деятельности официально закреплены за понятием «информатика»?
3. Какие сферы человеческой деятельности и в какой степени затрагивает информатика?
4. Назовите основные составные части информатики и основные направления ее применения.
5. Что подразумевается под понятием «информация» в бытовом, естественно-научном и техническом смыслах?
6. Что необходимо добавить в систему «источник информации – приемник информации», чтобы осуществлять передачу сообщений?

7. Приведите примеры ситуаций, в которых информация:

- |                      |                    |                 |
|----------------------|--------------------|-----------------|
| а) создается;        | д) копируется;     | и) передается;  |
| б) обрабатывается;   | е) воспринимается; | к) разрушается; |
| в) запоминается;     | ж) измеряется;     | л) ищется;      |
| г) делится на части; | з) принимается;    | м) упрощается.  |

8. Приведите примеры информации:

- а) достоверной и недостоверной;
- б) полной и неполной;
- в) ценной и малоценной;
- г) своевременной и несвоевременной;
- д) понятной и непонятной;
- е) доступной и недоступной для усвоения;
- ж) краткой и пространной.

9. Назовите системы сбора и обработки информации в теле человека.

10. Приведите примеры технических устройств и систем, предназначенных для сбора и обработки информации.

11. От чего зависит информативность сообщения, принимаемого человеком?

12. Почему количество информации в сообщении удобнее оценивать не по степени увеличения знания об объекте, а по степени уменьшения неопределенности наших знаний о нем?

13. Как определяется единица измерения количества информации?

14. В каких случаях и по какой формуле можно вычислить количество информации, содержащейся в сообщении?

15. Почему в формуле Хартли за основание логарифма взято число 2?

16. При каком условии формула Шеннона переходит в формулу Хартли?

17. Что определяет термин «бит» в теории информации и в вычислительной технике?

18. Приведите примеры сообщений, информативность которых можно определить однозначно.

19. Приведите примеры сообщений, содержащих один (два, три) бит (-а) информации.

## Глава 2. КОДИРОВАНИЕ ЧИСЛОВОЙ И СИМВОЛЬНОЙ ИНФОРМАЦИИ, КОДОВЫЕ ТАБЛИЦЫ

### 2.1. Кодирование данных двоичным кодом

Для автоматизации работы с данными, относящимися к различным типам, очень важно унифицировать их форму представления. Для этого обычно используется прием кодирования, т. е. выражение данных одного типа через данные другого типа. Естественные человеческие языки – это не что иное, как системы кодирования понятий для выражения мыслей посредством речи. К языкам близко примыкают азбуки (системы кодирования компонентов языка с помощью графических символов).

Своя система кодирования существует в вычислительной технике. Она называется двоичным кодированием и основана на представлении данных последовательностью всего двух знаков: 0 и 1 – битов (см. 1.2).

Одним битом могут быть выражены два понятия: 0 или 1 (да или истина, черное или белое, истина или ложь и т. п.).

Если количество битов увеличить до двух, то уже можно выразить четыре различных понятия:

00 01 10 11

Тремя битами можно закодировать восемь различных значений:

000 001 010 011 100 101 110 111

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

$$N = 2^m,$$

где  $N$  – количество независимых кодируемых значений;

$m$  – разрядность двоичного кодирования, принятая в данной системе.

#### *2.1.1. Кодирование целых и действительных чисел, текстовой информации*

Для кодирования целых чисел от 0 до 255 достаточно иметь 8 разрядов двоичного кода (8 бит). Шестнадцать бит позволяют закодировать целые числа от 0 до 65 535, а 24 бита – уже более 16,5 миллионов разных значений.



Для кодирования действительных чисел используют 80-разрядное кодирование. При этом число предварительно преобразуется в *нормализованную форму*:

$$3,1415926 = 0,31415926 \cdot 10^1$$

$$300\,000 = 0,3 \cdot 10^6$$

$$123\,456\,789 = 0,123456789 \cdot 10^{10}$$

Первая часть числа называется мантиссой, а вторая – характеристикой. Большую часть из 80 бит отводят для хранения мантиссы (вместе со знаком) и некоторое фиксированное количество разрядов отводят для хранения характеристики (тоже со знаком).

Если каждому символу алфавита сопоставить определенное целое число (например, порядковый номер), то с помощью двоичного кода можно кодировать и текстовую информацию. Для кодирования 256 различных символов достаточно восьми двоичных разрядов или 1 байт информации. Этого хватит, чтобы выразить различными комбинациями восьми битов все символы английского и русского языков, как строчные, так и прописные, а также знаки препинания, символы основных арифметических действий и некоторые общепринятые специальные символы, например символ «§».

Для того чтобы весь мир одинаково кодировал текстовые данные, нужны единые таблицы кодирования, а это пока невозможно из-за противоречий между символами национальных алфавитов, а также противоречий корпоративного характера.

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

Аналогичные системы кодирования текстовых данных были разработаны и в других странах. Отсутствие единого стандарта в этой области привело к множественности одновременно действующих кодировок. Только в России можно указать три действующих стандарта кодировки и еще два устаревших.

Например, кодировка символов русского языка, известная как кодировка *Windows-1251* используется на большинстве локальных компьютеров, работающих на платформе Windows.

Другая распространенная кодировка носит название КОИ-8 (код обмена информацией, восьмизначный). Ее происхождение относится ко временам действия Совета Экономической Взаимопомощи государств Восточной Европы. Сегодня кодировка КОИ-8 имеет широкое распространение в компьютерных сетях на территории России и в российском секторе Интернета.

Международный стандарт, в котором предусмотрена кодировка символов русского алфавита, носит название кодировки ISO (*International Standard Organization* – Международный институт стандартизации). На практике данная кодировка используется редко.

В связи с изобилием систем кодирования текстовых данных, действующих в России, была принята для практического применения Универсальная система кодирования текстовых данных UNICODE. Эта система, основана на 16-разрядном кодировании символов. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65 536 различных символов – этого поля достаточно для размещения в одной таблице символов большинства языков планеты. Эту кодировку поддерживает продукты Microsoft, начиная с Microsoft Office 97.

Для характеристики информации используется показатель «мощность алфавита», вычисляемый количеством информации, приходящимся на один знак алфавита (количеством бит, требующимся для кодировки одного символа).

*Пример.* Определите мощность алфавита, с помощью которого записано сообщение, содержащее 8182 символа, если его объем составляет 5 Кбайт.

*Решение.* Переведите объем сообщения в биты:

$$1 \text{ Кбайт} = 1024 \text{ байт. } 5 \text{ Кбайт} = 1024 \cdot 5 = 5120 \text{ байт.}$$

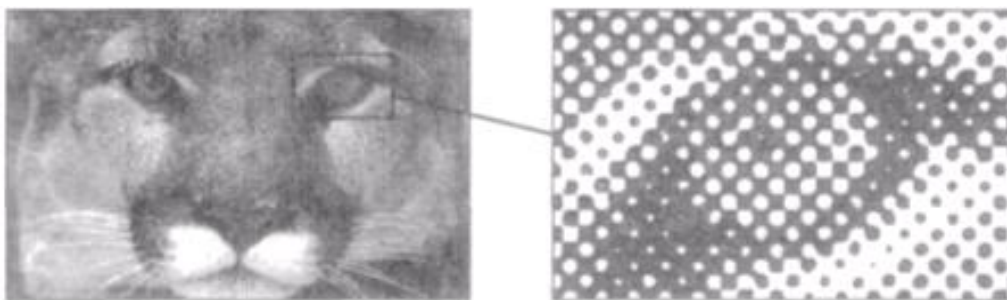
Для перевода в биты необходимо 5120 байт умножить на 8 (напомним, что 1 байт = 8 бит). Получим 40960 бит.

Количество бит, приходящееся на один символ алфавита =  $40960 / 8182 = 5$ .

Количество символов алфавита  $N = 2^m = N = 2^5 = 32$ . Мощность алфавита составит 5 бит.

### **2.1.2. Кодирование графических данных**

Если рассмотреть с помощью увеличительного стекла черно-белое графическое изображение, напечатанное в газете или книге, то можно увидеть, что оно состоит из мельчайших точек (пикселей), образующих характерный узор, называемый растром (рис. 2.1).



*Рисунок 2.1 – Растр – метод кодирования графической информации, издавна принятый в полиграфии*

Черно-белое изображение без градаций серого представляется в виде одного бита.

Общепринятым на сегодняшний день считается представление черно-белых иллюстраций в виде комбинации точек с 256 градациями серого цвета. Таким образом, для кодирования яркости любой точки обычно достаточно восьмиразрядного двоичного числа.

Для кодирования цветных графических изображений применяется принцип декомпозиции произвольного цвета на основные составляющие. В качестве таких составляющих используют три основных цвета: красный (Red – R), зеленый (Green – G) и синий (Blue – B). Такая система кодирования называется системой RGB по первым буквам названий основных цветов.

Если для кодирования яркости каждой из основных составляющих использовать по 256 значений (восемь двоичных разрядов), как это принято для полутоновых черно-белых изображений, то на кодирование цвета одной точки надо затратить 24 разряда. При этом система кодирования обеспечивает однозначное определение 16,5 млн различных цветов, что на самом деле близко к чувствительности человеческого глаза. Режим представления цветной графики с использованием 24 двоичных разрядов называется (True Color).

Каждому из основных цветов можно поставить в соответствие дополнительный цвет, т. е. цвет, дополняющий основной цвет до белого. Соответственно, дополнительными цветами являются: голубой (Cyan – C), пурпурный (Magenta – M) и желтый (Yellow – Y). В полиграфии используется еще и четвертая краска – черная (Black – K). Поэтому данная система кодирования обозначается четырьмя буквами CMYK (черный цвет обозначается буквой K, потому что буква B уже занята синим цветом), и для представления цветной графики в

этой системе надо иметь 32 двоичных разряда. Такой режим тоже называется полноцветным (True Color).

Кодирование цветной графики 16-разрядными двоичными числами для сокращения объема данных (но при этом диапазон кодируемых цветов заметно сокращается) называется режимом High Color.

Наиболее распространенной разрешающей способностью экрана является разрешение 800 на 600 точек, т. е. 480000 точек. Рассчитаем необходимый для режима true color объем видеопамати:  $1 = 2$  байт  $480\ 000 = 960\ 000$  байт = 937,5 Кб. Аналогично рассчитывается объем видеопамати, необходимый для хранения битовой карты изображений при других видеорежимах.

Разрешение	16 цветов	256 цветов	65536 цветов
640 × 480	150 Кб	300 Кб	600 Кб
800 × 600	234,4 Кб	468,8 Кб	937,5 Кб
1024 × 768	384 Кб	768 Кб	1,5 Мб
1280 × 1024	640 Кб	1,25 Мб	2,5 Мб

Кроме растрового изображения, компьютеры могут создавать и векторные изображения, которые строятся на основе математических описаний прямых и кривых линиях – векторах.



*Рисунок 2.2 – Векторное изображение*

Нарисованную таким образом фигуру можно переместить, перевернуть, увеличить или уменьшить как независимый объект, поскольку программа сохраняет описание параметров фигур не в графическом представлении, а в виде математических формул. Это свойство делает векторную графику удобной для создания иллюстраций, шрифтовых заставок, логотипов и других объектов с четкими и ясными контурами.

### ***2.1.3. Кодирование звуковой информации***

Приемы и методы работы со звуковой информацией пришли в вычислительную технику поздно. В итоге методы кодирования звуковой информации двоичным кодом далеки от стандартизации. Можно выделить два основных направления.

Метод FM (Frequency Modulation) основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических сигналов разных частот, каждый представляет правильную синусоиду, а следовательно, может быть описан числовыми параметрами, т. е. кодом. В природе звуковые сигналы имеют непрерывный спектр, т. е. являются аналоговыми. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства – аналогово-цифровые преобразователи (АЦП). Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют цифро-аналоговые преобразователи (ЦАП). При таких преобразованиях неизбежны потери информации, связанные с методом кодирования, поэтому качество звукозаписи обычно получается не вполне удовлетворительным и соответствует качеству звучания простейших электромузыкальных инструментов с окрасом, характерным для электронной музыки. В то же время данный метод кодирования обеспечивает весьма компактный код, потому он нашел применение еще в те годы, когда ресурсы средств вычислительной техники были явно недостаточны.

Метод таблично-волнового (Wave-Table) синтеза. В заранее подготовленных таблицах хранятся образцы звуков. В технике такие образцы называют сэмплами. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звука. Поскольку в качестве образцов используют реальные звуки, то качество звука, полученного в результате синтеза, получается очень высоким и приближается к качеству звучания музыкальных инструментов.

## Вопросы для повторения и самоконтроля

1. Во сколько раз нужно увеличить количество разрядов в системе двоичного кодирования для увеличения в два раза количества значений, которое может быть выражено в данной системе?
2. Какое количество целых чисел позволит закодировать восьмиразрядное двоичное число?
3. Какое количество целых чисел позволит закодировать шестнадцатиразрядное двоичное число?
4. Что такое нормализованная форма числа?
5. Что такое мантисса и характеристика нормализованного числа?
6. Каким способом кодируются символы алфавита?
7. Как называется система кодирования текстовых данных, принятых в англоязычных странах?
8. Чем отличаются системы кодирования текстовых данных Windows-1251, КОИ-8, ISO, UNICODE?
9. Какое свойство характеризует показатель «мощность алфавита»?
10. Назовите два основных способа представления графических изображений.
11. Какая связь между пикселем и растром?
12. Сколькими разрядами представляется черно-белое изображение без градации серого цвета?
13. Сколькими разрядами представляется черно-белое изображение с градацией серого цвета?
14. Какие три основных цвета применяют для кодирования графических изображений?
15. Сколько разрядов двоичных чисел надо использовать для представления графической информации в режиме True Color?
16. Какие дополнительные цвета применяют для кодирования графических изображений?
17. Сколько разрядов двоичных чисел надо использовать для представления графической информации в режиме полноцветного True Color?
18. Сколько разрядов двоичных чисел надо использовать для представления графической информации в режиме High Color?
19. Что такое векторное изображение?
20. Какие два основных метода кодирования звуковой информации вам известны? В чем их различие?

## Задания для самостоятельной работы

1. Сколько Кб составляет сообщение, содержащее 12288 битов?
2. Письмо занимает 2 страницы по 25 строк. В каждой строке записано по 40 символов. Каков объем информации в письме?
3. Запишите множество вариантов загорания двух светофоров, расположенных на соседних перекрестках.
4. Три человека, Иванов, Петров и Сидоров, образуют очередь. Запишите все возможные варианты образования этой очереди.
5. Назовите все возможные комбинации из двух различных нот. Всего нот семь: до, ре, ми, фа, соль, ля, си.
6. Пусть голосуют 3 человека (голосование да/нет). Запишите все возможные исходы голосования.
7. Предположим, что имеются 3 автомобильные дороги, идущие от Красноярска до Ачинска, и 4 – от Ачинска до Минусинска. Сколькими способами можно выбрать дорогу от Красноярска в Минусинска через Ачинск? Попытайтесь найти систематический метод для последовательного нахождения решения так, чтобы можно было составить список способов, не пропустив ни одного из них.
8. Поезд находится на одном из восьми путей. Сколько бит информации содержит сообщение о том, где находится поезд?
9. Сколько существует различных двоичных последовательностей из одного, двух, трех, четырех, восьми символов?
10. Каков информационный объем сообщения «Я помню чудное мгновенье» при условии, что один символ кодируется одним байтом и соседние слова разделены одним пробелом?
11. Сколько бит необходимо, чтобы закодировать следующие оценки: неудовлетворительно, удовлетворительно, хорошо и отлично?
12. Сколько различных символов, закодированных байтами, содержится в сообщении:  
1101001100011100110100110001110001010111?
13. Сколько байт памяти необходимо, чтобы закодировать изображение на экране компьютерного монитора, который может отображать 1280 точек по горизонтали и 1024 точек по вертикали при 256 цветах?
14. В процессе преобразования графического файла количество цветов уменьшилось с 4096 до 64. Во сколько раз уменьшился размер файла?
15. Дисплей имеет разрешающую способность в графическом режиме  $640 \times 400$ , а в текстовом – 16 строк по 80 позиций в строке. Какова разрешающая способность одной текстовой позиции в пикселях?

16. Решите уравнение:  $8^x$  (бит) = 32 (Кбайт).

17. Определите правила формирования приведенных ниже последовательностей и вставьте пропущенные числа:

- |                        |                             |                 |
|------------------------|-----------------------------|-----------------|
| а) 1, 3, 5, ..., 9;    | ж) 128, 64, 32, ..., 8;     | н) 15 (27) 42   |
| б) 20, 15, ..., 5;     | з) 4, 9, 17, 35, ..., 139;  | 30 (...) 55;    |
| в) 1, 2, 4, ..., 16;   | и) 1, 2, 2, 4, 8, ..., 256; | о) 10 (50) 15   |
| г) 1, 4, 9, ..., 25;   | к) 2, 3, 10, 15, ..., 35;   | 17 (...) 20;    |
| д) 1, 8, 27, ..., 125; | л) 1, 3, 3, 9, ..., 6561;   | п) 143 (56) 255 |
| е) 1, 2, 6, ..., 120;  | м) к, о, ж, з, г, ..., ф;   | 218 (...) 114.  |

18. В режиме True Color на хранение кода каждого пикселя отводится:

- 16 бит;
- 16 байт;
- 24 бита.

19. Минимальной единицей измерения графического изображения на экране монитора является:

- mm;
- sm;
- pixel;
- inch.

20. Растровый графический файл содержит черно-белое изображение (без градаций серого цвета) размером  $100 \times 100$  точек. Объем памяти, требуемый для хранения этого файла:

- 1000 бит;
- 10000 бит;
- 10000 байт.

21. Растровый файл, содержащий черно-белый (без оттенков серого) квадратный рисунок, имеет объем 200 байт. Рассчитайте размер стороны квадрата (в пикселях):

- 15;
- 40;
- 1000.

22. Объем изображения, размером  $40 \times 50$  пикселей, составляет 2000 байт. Изображение использует:

- 8 цветов;
- 256 цветов;
- 16777216 цветов.



23. Известно, что видеопамять компьютера имеет объем 512 Кбайт. Разрешающая способность экрана 640 на 200 пикселей. Количество страниц экрана, которое разместится одновременно в видеопамяти при палитре:

- из 8 цветов;
- 16 цветов;
- 256 цветов?

*Ответы к заданиям для самостоятельной работы*

1.  $12288 \text{ бит} / 8 = 1536 \text{ байт} = 1536 \text{ байт} / 1024 = 1,5 \text{ Кбайт}$ .

2. 2 страницы по 25 строк. В каждой строке записано по 40 символов. Тогда  $2 \cdot 25 \cdot 40 = 2000$  символов. Примем 1 байтовую систему кодирования, тогда объем информации составит 2000 байт.

3. Обозначим номера светофоров цифрами 1 и 2, а цвета их загорания – буквами З (зеленый), Ж (желтый) и К (красный). Тогда искомое множество содержит следующие 9 элементов:  $Z_1-Z_2$ ,  $Z_1-Z_2$ ,  $Z_1-K_2$ ,  $Z_1-K_2$ ,  $Z_1-K_2$ ,  $Z_1-K_2$ ,  $Z_1-K_2$ ,  $Z_1-K_2$ ,  $Z_1-K_2$ .

4. Вариантов всего шесть:

- |                             |                             |
|-----------------------------|-----------------------------|
| 1) Иванов, Петров, Сидоров; | 4) Петров, Сидоров, Иванов; |
| 2) Иванов, Сидоров, Петров; | 5) Сидоров, Иванов, Петров; |
| 3) Петров, Иванов, Сидоров; | 6) Сидоров, Петров, Иванов. |

5. Комбинаций всего 42:

до-ре, до-ми, ..., до-си, ре-до, ре-ми, ..., ре-си, ми-до, ми-ре, ..., ми-си, ..., си-до, си-ре, ..., си-ля.

6. Исходов всего восемь:

да-да-да, да-да-нет, да-нет-да, да-нет-нет, нет-да-да, нет-да-нет, нет-нет-да, нет-нет-нет.

7. Двенадцатью способами.

8. 3 бита информации ( $8 = 2^3$ ).

9. Двоичных последовательностей из одного бита всего  $2^1 = 2$ , из двух битов –  $2^2 = 4$ , из трех битов –  $2^3 = 8$ , из четырех битов –  $2^4 = 16$ , из восьми битов –  $2^8 = 256$ .

10. 24 сивола = 24 байта, или 192 бита.

11. Два бита: 00 – неудовлетворительно, 01 – удовлетворительно, 10 – хорошо, 11 – отлично.

12. Разбиваем сообщение на восьмерки битов (то есть, на байты):

01001100 01110011 01001100 01110011 01010111.

Сравнивая байты между собой, видим, что первый и третий, а также второй и четвертый байты одинаковые. Следовательно, различных символов всего три.

13. Всего на экране монитора  $1280 \times 1024 = 1310720$  точек. Для кодирования каждой из точек, которые могут быть окрашены в 256 цветов ( $256 = 2^8$ ) требуется 8 бит или 1 байт. Таким образом, для кодирования всего изображения требуется  $1310720$  байт  $= 1,25 \cdot 2^{20}$  байт  $= 1,25$  Мбайт.

14.  $4096 = 2^{12}$ .  $64 = 2^6$ . Тогда  $12 / 6 = 2$  раза.

15.  $640 \cdot 400 = 256000$  пикселей.  $16 \cdot 80 = 1280$  символов.  $256000 / 1280 = 200$  пикселей на знак.

16. Выравниваем размерности в левой и правой частях уравнения с учетом того, что  $1 \text{ Кбайт} = 2^{10}$  бит. Затем приводим обе части к одному основанию 2. Имеем:  $2^{3x} = 2^5 \cdot 2^{10}$  или  $2^{3x} = 2^{15}$ . Переходим к равносильному уравнению  $3x = 15$ , откуда  $x = 15:3 = 5$ .

17 а) 7 (чтобы получить следующее число, нужно к предыдущему прибавить 2:  $a_1 = 1$ ,  $a_i = a_{i-1} + 2$ ,  $i = 2, 3, \dots$ );

б) 10 (чтобы получить следующее число, нужно от предыдущего отнять 5:  $a_1 = 20$ ,  $a_i = a_{i-1} - 5$ ,  $i = 2, 3, \dots$ );

в) 8 (чтобы получить следующее число, нужно предыдущее умножить на 2:  $a_1 = 1$ ,  $a_i = a_{i-1} \cdot 2$ ,  $i = 2, 3, \dots$ );

г) 16 (возвести в квадрат числа 1, 2, 3, ... :  $a_i = i^2$ ,  $i = 1, 2, 3, \dots$ );

д) 64 (возвести в куб числа 1, 2, 3, ... :  $a_i = i^3$ ,  $i = 1, 2, 3, \dots$ );

е) 24 (чтобы получить очередное число, нужно предыдущее умножить на номер числа:  $a_1 = 1$ ,  $a_i = a_{i-1} \cdot i$ ,  $i = 2, 3, \dots$ );

ж) 8 (чтобы получить следующее число, нужно предыдущее разделить на 2:  $a_1 = 128$ ,  $a_i = a_{i-1} : 2$ ,  $i = 2, 3, \dots$ );

з) 69 (чтобы получить следующее число, нужно предыдущее умножить на 2 и к полученному произведению поочередно прибавлять и вычитать единицу:  $a_1 = 4$ ,  $a_i = 2a_{i-1} + (-1)^i$ ,  $i = 2, 3, \dots$ );

и) 32 ( $a_1 = 1$ ,  $a_2 = 2$ ,  $a_i = a_{i-1} \cdot a_{i-2}$ ,  $i = 3, 4, \dots$ );

к) 26 ( $a_1 = 2$ ,  $a_i = i^2 + (-1)^{i-1}$ ,  $i = 2, 3, \dots$ );

л) 81 ( $a_1 = 1$ ,  $a_2 = 3$ ,  $a_i = a_1 \cdot a_2 \cdot \dots \cdot a_{i-1}$ ,  $i = 3, 4, \dots$ )

м) с (выписаны первые буквы цветов радуги: с – «синий»);

н) 25 (число в скобках есть разность между числами вне скобок);

о) 74 (удвоенная сумма чисел, стоящих вне скобок);

п) 52 (полуразность чисел, стоящих вне скобок)

18. 24 бита.

19. Pixel.

20. 10000 бит.

21. 40 пикселей.  $40 \cdot 40 = 1600$  бит.  $200$  байт  $= 200 \cdot 8 = 1600$ .

22. 256 цветов. Количество пикселей в изображении составит  $40 \cdot 50 = 2000$  пикселей. На 2000 пикселей приходится 2000 байт. На один пиксель приходится один байт или восемь разрядов. Одним байтом можно описать  $2^8 = 256$  цветов.

23. 512 Кбайт =  $512 \cdot 1024 = 524288$  байт.  $640 \cdot 200 = 128000$  пикселей.

При 8 цветах ( $2^3$ ) 128000 пикселей займут объем  $128000 \cdot 3 / 8 = 48000$  байт.

При 16 цветах ( $2^4$ ) 128000 пикселей займут объем  $128000 \cdot 4 / 8 = 64000$  байт.

При 256 цветах ( $2^8$ ) 128000 пикселей займут объем  $128000 \cdot 8 / 8 = 128000$  байт.

Тогда  $524288 \text{ байт} / 48000 \text{ байт} = 10,92$  страницы.

$524288 \text{ байт} / 64000 \text{ байт} = 8,19$  страницы.

$524288 \text{ байт} / 128000 \text{ байт} = 4,01$  страницы.

## Глава 3. ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ И РАБОТЫ КОМПЬЮТЕРОВ

### 3.1. Основные понятия о компьютере

Компьютер (англ. *computer* – вычислитель) представляет собой программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами [2, 13].

Существует два основных класса компьютеров:

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

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

Основу компьютеров образует аппаратура (*HardWare*), построенная в основном с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (*SoftWare*) – заранее заданных, четко определенных последовательностей арифметических, логических и других операций.

Любая компьютерная программа представляет собой последовательность отдельных команд.

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

Совокупность команд, выполняемых данным компьютером, называется системой команд этого компьютера.

### 3.2. Устройство компьютера

Структура компьютера основана на общих логических принципах, позволяющих выделить следующие главные устройства:

- память (запоминающее устройство, ЗУ), состоящую из перенумерованных ячеек;
- процессор, включающий в себя устройство управления (УУ) и арифметико-логическое устройство (АЛУ);
- устройство ввода;

– устройство вывода.

Эти устройства соединены каналами связи, по которым передается информация.

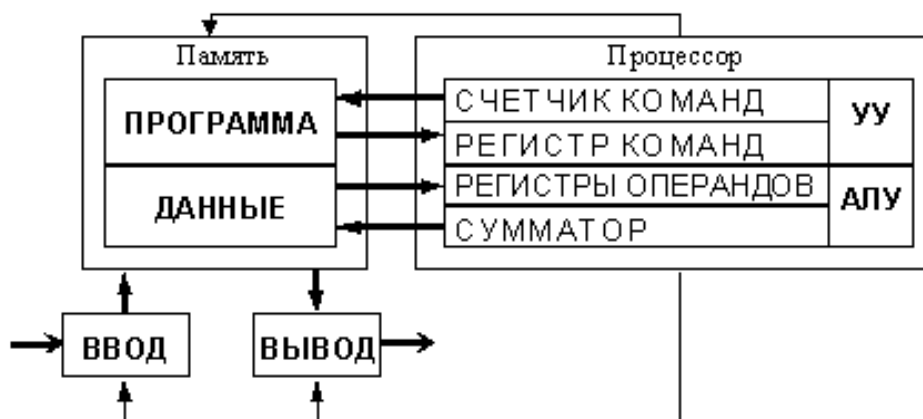


Рисунок 3.1 – Общая схема компьютера

Основные устройства компьютера и связи между ними представлены на схеме (рис. 3.1). Жирными стрелками показаны пути и направления движения информации, а простыми стрелками – пути и направления передачи управляющих сигналов.

Функции памяти:

- прием информации из других устройств;
- запоминание информации;
- выдача информации по запросу в другие устройства машины.

Функции процессора:

- обработка данных по заданной программе путем выполнения арифметических и логических операций;
- программное управление работой устройств компьютера.

Та часть процессора, которая выполняет команды, называется арифметико-логическим устройством (АЛУ), а другая его часть, выполняющая функции управления устройствами, называется устройством управления (УУ). Обычно эти два устройства выделяются чисто условно, конструктивно они не разделены.

В составе процессора имеется ряд специализированных дополнительных ячеек памяти, называемых регистрами.

Регистр выполняет функцию кратковременного хранения числа или команды. Над содержимым некоторых регистров специальные электронные схемы могут выполнять некоторые манипуляции. Например, «вырезать» отдельные части команды для последующего их

использования или выполнять определенные арифметические операции над числами.

Основным элементом регистра является электронная схема, называемая триггером, которая способна хранить одну двоичную цифру (разряд двоичного кода).

Регистр представляет собой совокупность триггеров, связанных друг с другом определенным образом общей системой управления.

Существует несколько типов регистров, отличающихся видом выполняемых операций. Некоторые важные регистры имеют свои названия, например:

- сумматор – регистр АЛУ, участвующий в выполнении каждой операции;

- счетчик команд – регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки программы из последовательных ячеек памяти;

- регистр команд – регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения. Часть его разрядов используется для хранения кода операции, остальные – для хранения кодов адресов операндов.

### 3.3. Принципы построения компьютера

В основу построения подавляющего большинства компьютеров положены следующие общие принципы, сформулированные в 1945 г. американским ученым Джоном фон Нейманом.

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

Выборка программы из памяти осуществляется с помощью счетчика команд. Этот регистр процессора *последовательно увеличивает хранимый в нем адрес очередной команды на длину команды.*

А так как команды программы расположены в памяти друг за другом, то тем самым организуется выборка цепочки команд из последовательно расположенных ячеек памяти.

Если же нужно после выполнения команды перейти не к следующей, а к какой-то другой команде, используют команды условного или безусловного переходов, которые *вносят в счетчик команд номер ячейки памяти, содержащей следующую команду.* Выборка команд из памяти прекращается после достижения и выполнения команды «стоп».

Таким образом, *процессор исполняет программу автоматически, без вмешательства человека.*

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

3. Принцип адресности. *Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.* Отсюда следует возможность давать имена областям памяти, так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен.

Компьютеры, построенные на этих принципах, относятся к типу фон-неймановских. Но существуют компьютеры, принципиально отличающиеся от фон-неймановских. Для них, например, может не выполняться принцип программного управления, т. е. они могут работать без «счетчика команд», указывающего текущую выполняемую команду программы. Для обращения к какой-либо переменной, хранящейся в памяти, этим компьютерам не обязательно давать ей имя. Такие компьютеры называются не-фон-неймановскими.

Реальная структура ЭВМ значительно сложнее, чем рассмотренная выше, за счет включения в нее дополнений, направленных на повышение производительности и приближение функциональных возможностей ЭВМ к потребностям пользователей. В современных ПК все чаще осуществляется отход от традиционной архитектуры фон Неймана. Однако во многом структура ЭВМ и принципы ее построения и функционирования сохраняются.

### 3.4. Команда компьютера

В общем случае команда содержит следующую информацию:

- код выполняемой операции;
- указания по определению операндов (или их адресов);
- указания по размещению получаемого результата.

В зависимости от количества операндов команды бывают:

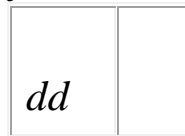
- одноадресные;
- двухадресные;
- трехадресные;
- переменнадресные.

Команды хранятся в ячейках памяти в двоичном коде.

В современных компьютерах длина команд переменная (обычно от двух до четырех байтов), а способы указания адресов переменных весьма разнообразные.

Рассмотрим несколько возможных вариантов команды сложения (англ. *add* – сложение), при этом вместо цифровых кодов и адресов будем пользоваться условными обозначениями:

- одноадресная команда *add x* (содержимое ячейки *x* сложить с содержимым сумматора, а результат оставить в сумматоре)



- двухадресная команда *add x, y* (сложить содержимое ячеек *x* и *y*, а результат поместить в ячейку *y*)



- трехадресная команда *add x, y, z* (содержимое ячейки *x* сложить с содержимым ячейки *y*, сумму поместить в ячейку *z*)



Выполнение команды можно проследить по схеме (см. рис. 3.1).

Как правило, этот процесс разбивается на следующие этапы:

- из ячейки памяти, адрес которой хранится в счетчике команд, выбирается очередная команда; содержимое счетчика команд при этом увеличивается на длину команды;



- выбранная команда передается в устройство управления на регистр команд;
- устройство управления расшифровывает адресное поле команды;
- по сигналам УУ операнды считываются из памяти и записываются в АЛУ на специальные регистры операндов;
- УУ расшифровывает код операции и выдает в АЛУ сигнал выполнить соответствующую операцию над данными;
- результат операции либо остается в процессоре, либо отправляется в память, если в команде был указан адрес результата;
- все предыдущие этапы повторяются до достижения команды «СТОП».

### **3.5. Архитектура и структура компьютера**

При рассмотрении компьютерных устройств принято различать их архитектуру и структуру.

Архитектурой компьютера называется его описание на некотором общем уровне, включающее описание пользовательских возможностей программирования, системы команд, системы адресации, организации памяти и т. д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного ЗУ, внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их совместимость с точки зрения пользователя.

Структура компьютера – это совокупность его функциональных элементов и связей между ними. Элементами могут быть самые различные устройства – от основных логических узлов компьютера до простейших схем. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации.

Наиболее распространены следующие архитектурные решения.

1. Классическая архитектура (архитектура фон Неймана) – одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд – программа (см. рис. 3.1). Это однопроцессорный компьютер. К этому типу архитектуры относится и архитектура персонального компьютера с общей шиной (рис. 3.2).

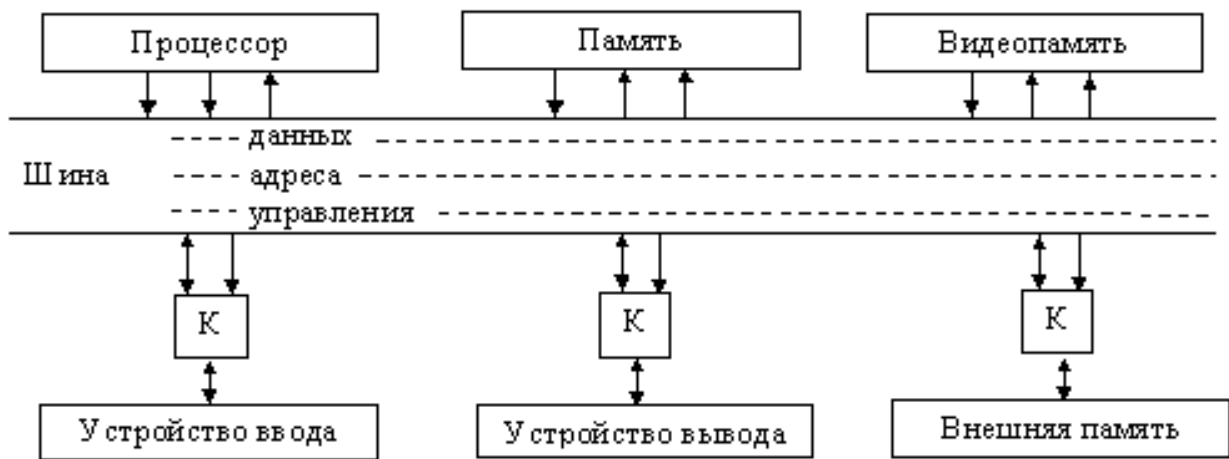


Рисунок 3.2 – Магистрально-модульный принцип организации компьютера

Все функциональные блоки здесь связаны между собой общей шиной, называемой также системной магистралью.

Физически магистраль представляет собой многопроводную линию с гнездами для подключения электронных схем. Совокупность проводов магистралей разделяется на отдельные группы: шину адреса, шину данных и шину управления.

Периферийные устройства (принтер и др.) подключаются к аппаратуре компьютера через специальные контроллеры – устройства управления периферийными устройствами. Они связывают периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

2. Многопроцессорная архитектура. Наличие в компьютере нескольких процессоров означает, что параллельно может быть организовано много потоков данных и много потоков команд. Таким образом, параллельно могут выполняться несколько фрагментов одной задачи. Структура такой машины, имеющей общую оперативную память и несколько процессоров, представлена на рисунке 3.3.

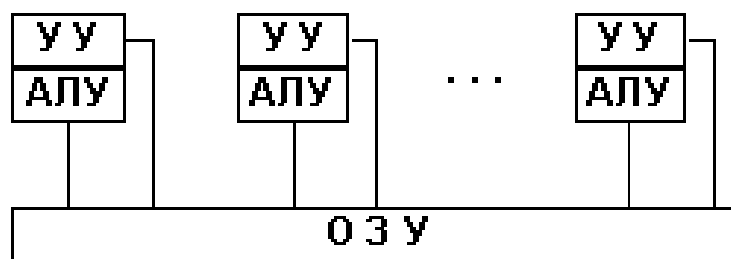


Рисунок 3.3 – Архитектура многопроцессорного компьютера

3. Многомашинная вычислительная система. Здесь *несколько процессоров, входящих в вычислительную систему, не имеют общей оперативной памяти, а имеют каждый свою (локальную)*. Каждый компьютер в многомашинной системе имеет классическую архитектуру, и такая система применяется достаточно широко. Однако эффект от применения такой вычислительной системы может быть получен только при решении задач, имеющих очень специальную структуру: *она должна разбиваться на столько слабо связанных подзадач, сколько компьютеров в системе*.

Преимущество в быстродействии многопроцессорных и многомашинных вычислительных систем перед однопроцессорными очевидно.

4. Архитектура с параллельными процессорами. Здесь *несколько АЛУ работают под управлением одного УУ*. Это означает, что множество данных может обрабатываться по одной программе, т. е. по одному потоку команд. Высокое быстродействие такой архитектуры можно получить только на задачах, в которых одинаковые вычислительные операции выполняются одновременно на различных однотипных наборах данных. Структура таких компьютеров представлена на рисунке 3.4.

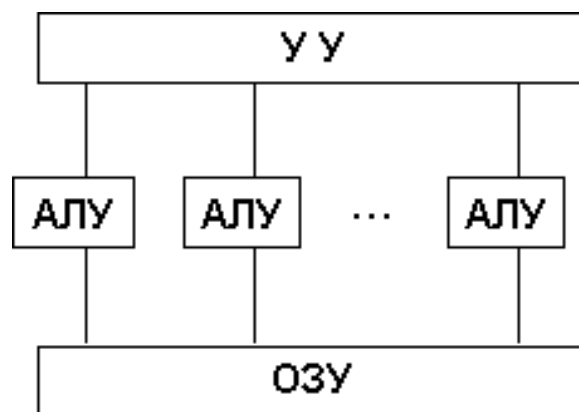


Рисунок 3.4 – Архитектура с параллельным процессором

В современных машинах часто присутствуют элементы различных типов архитектурных решений. Существуют и такие архитектурные решения, которые радикально отличаются от рассмотренных выше.

### 3.6. Устройство памяти компьютера

Память компьютера построена из двоичных запоминающих элементов – битов, объединенных в группы по 8 битов, которые называются байтами. (Единицы измерения памяти совпадают с единицами измерения информации). Все байты пронумерованы. Номер байта называется его адресом.

Байты могут объединяться в ячейки, которые называются также словами. Для каждого компьютера характерна определенная длина слова – два, четыре или восемь байтов. Это не исключает использования ячеек памяти другой длины (например, полуслово, двойное слово). Как правило, в одном машинном слове может быть представлено либо одно целое число, либо одна команда. Однако допускаются переменные форматы представления информации. Разбиение памяти на слова для четырехбайтовых компьютеров представлено в таблице 3.1.

Таблица 3.1 – Разбиение памяти на слова для четырехбайтовых компьютеров

Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
ПОЛУСЛОВО		ПОЛУСЛОВО		ПОЛУСЛОВО		ПОЛУСЛОВО	
СЛОВО				СЛОВО			
ДВОЙНОЕ СЛОВО							

Современные компьютеры имеют много разнообразных запоминающих устройств, которые сильно отличаются между собой по назначению, временным характеристикам, объему хранимой информации и стоимости хранения одинакового объема информации. Различают два основных вида памяти – внутреннюю (оперативную – ОЗУ) и внешнюю (ВЗУ).

Оперативная память с произвольным доступом – это быстрое запоминающее устройство не очень большого объема, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Внешняя память (ВЗУ) предназначена для длительного хранения программ и данных, и целостность ее содержимого не зависит от того, включен или выключен компьютер. В отличие от оперативной па-

мяти, внешняя не имеет прямой связи с процессором. Информация от ВЗУ к процессору и наоборот циркулирует по следующей цепочке (рис. 3.5):



Рисунок 3.5 – Циркуляция информации от ВЗУ к процессору и обратно

Перечень устройств, принципы работы внутренней и внешней памяти будут подробно рассмотрены в следующей главе.

Далее перейдем к рассмотрению устройства современного компьютера на примере персонального.

### Вопросы для повторения и самоконтроля

1. Что подразумевается под термином «компьютер»?
2. На какие два класса делят все компьютеры? В чем их различие?
3. Как расшифровывается термин «HardWare»?
4. Как расшифровывается термин «SoftWare»?
5. Что такое система команд компьютера?
6. Перечислите основные устройства компьютера.
7. Попробуйте нарисовать общую схему компьютера.
8. Основные функции памяти компьютера.
9. Основные функции процессора компьютера.
10. Что такое процессор? Из каких основных устройств он состоит?
11. Каким образом связаны регистры и триггеры?
12. Специализированные регистры и их назначение?
13. Назовите основные принципы построения компьютера, сформулированные фон Нейманом.
14. Расскажите о принципе программного управления компьютером.
15. Расскажите о принципе однородности памяти компьютера.
16. Расскажите о принципе адресности памяти компьютера.
17. Существуют ли компьютеры, принципы работы которых отличаются от фон-неймановских?
18. Какая информация содержится в команде компьютера?
19. Опишите основные варианты выполнения операции в одно-, двух- и трехадресных компьютерах. В каких компьютерах (с одина-

ковой тактовой частотой) вычислительные операции выполняются быстрее и почему?

20. Опишите процесс выполнения команды процессора.

21. Что такое «архитектура» и «структура» компьютера?

22. Классическая архитектура компьютера.

23. Многопроцессорная архитектура компьютера.

24. Многомашинная архитектура компьютера.

25. Архитектура компьютера с параллельным процессором.

26. Что такое машинное слово?

27. Нарисуйте схему разбиения памяти на слова для четырех-байтового компьютера.

28. В чем различие оперативной и внешней памяти? Нарисуйте схему циркуляции информации от внешних запоминающих устройств к процессору и обратно.

## Глава 4. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

Структура аппаратного обеспечения современного ПК приведена на рисунке 4.1, а расположение основных устройств, входящих в состав ПК, на рисунке 4.2.

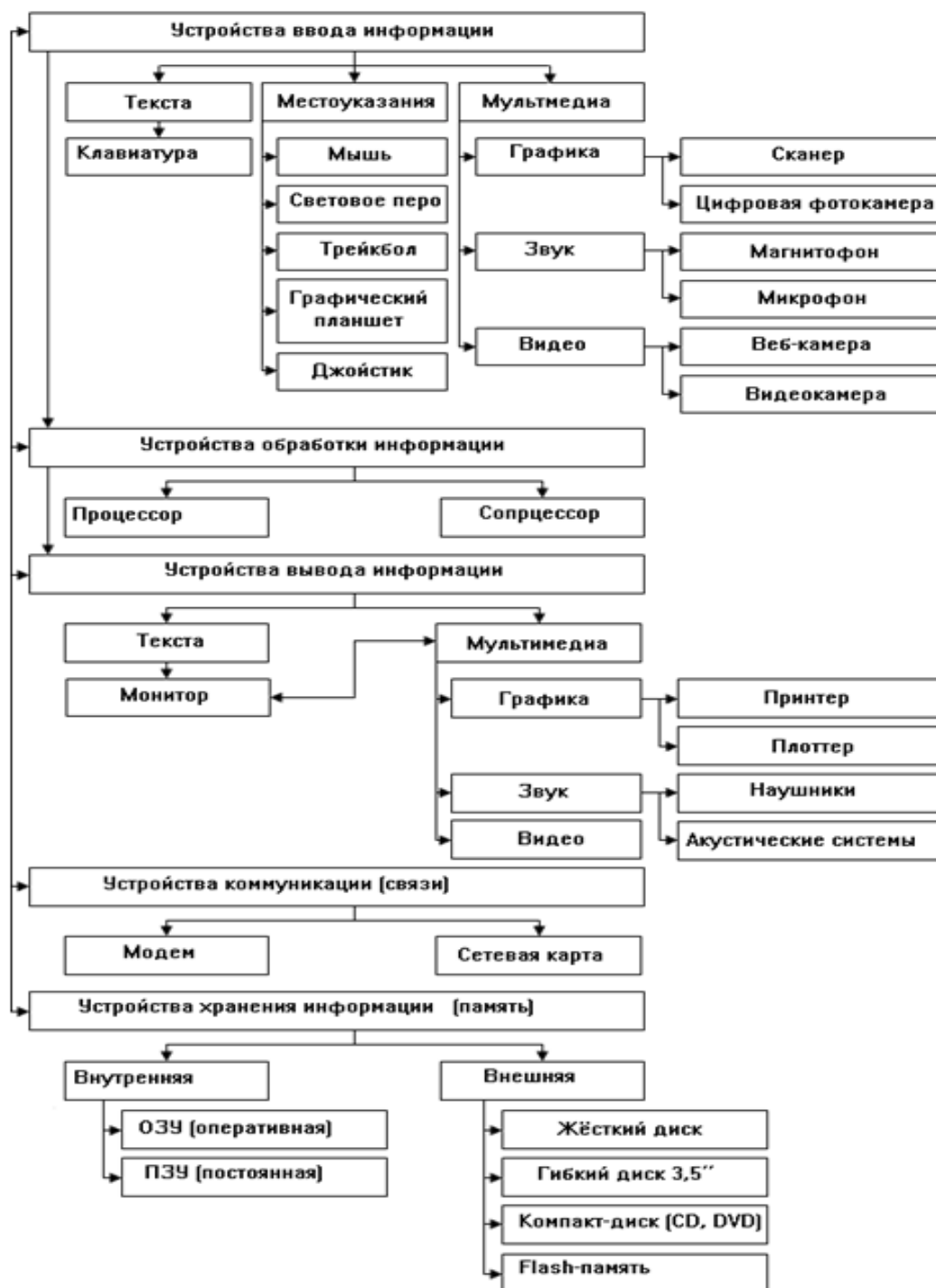
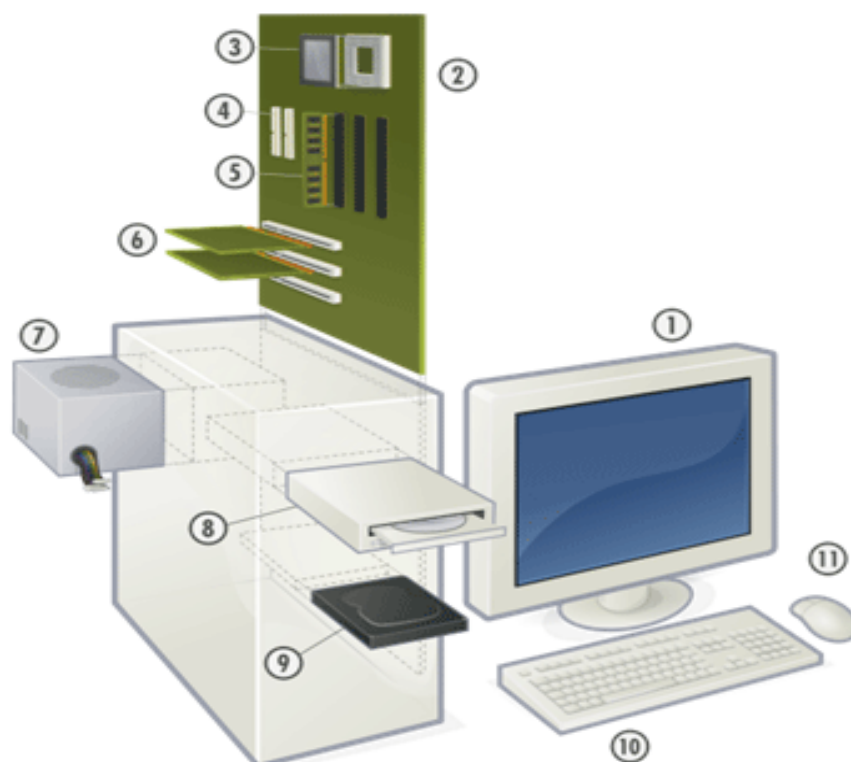


Рисунок 4.1 – Структура аппаратного обеспечения ПК



*Рисунок 4.2 – Расположение основных устройств, входящих в состав ПК:  
 1 – монитор; 2 – материнская плата; 3 – процессор; 4 – IDE-слот;  
 5 – оперативная память; 6 – платы расширения (видео, звуковая...);  
 7 – блок питания; 8 – привод для дисков (CD/DVD); 9 – винчестер;  
 10 – клавиатура; 11 – мышь*

Группировка устройств ПК представлена в таблице 4.1.

Таблица 4.1 – Группировка устройств персонального компьютера

Группа блоков и устройств	Блок и устройство
Основные блоки	Системный блок   монитор   устройства ввода-вывода
Устройства в составе системного блока	Материнская плата   центральный процессор   оперативная память   жесткий диск   графическая плата   звуковая плата   сетевая плата   дисковод   CD-привод   DVD-привод   TV-тюнер
Периферийные (внешние) устройства	Принтер   сканер   графопостроитель (плоттер)   модем   микрофон   акустика   ИБП – источник бесперебойного питания   клавиатура   мышь   графический планшет   тачпад   вебкамера   фотокамера   видекамера



## 4.1. Устройства, входящие в состав системного блока

### 4.1.1. Материнская плата

Материнская плата – печатная плата, на которой осуществляется монтаж большинства компонентов компьютерной системы. Название происходит от английского *motherboard*, иногда используется сокращение *МВ* или слово *mainboard* – главная плата.

Материнская плата обеспечивает связь между всеми устройствами ПК, посредством передачи сигнала от одного устройства к другому. Пример материнской платы для двухядерного процессора приведен на рис. 4.3.



Рисунок 4.3 – Материнская плата для двухядерного процессора

На поверхности материнской платы имеется большое количество разъемов, предназначенных для установки других устройств: *sockets* – гнезда для процессоров; *slots* – разъемы под оперативную память и платы расширения; контроллеры портов ввода/ вывода.

### 4.1.2. Центральный процессор

Центральный процессор, или центральное процессорное устройство (ЦПУ) (англ. *central processing unit* – CPU) – основная микросхема компьютера, в которой и производятся все вычисления.

Центральный процессор в общем случае содержит в себе (см. рис. 3.1 и 3.3):

- арифметико-логическое устройство;
- шины данных и шины адресов;

- регистры;
- счетчики команд;
- кэш – очень быструю память малого объема (от 8 до 512 Кбайт);
- математический сопроцессор чисел с плавающей точкой.

ЦПУ имеет размеры  $5 \times 5 \times 0,3$  см, устанавливается на материнской плате. На процессоре установлен большой радиатор, охлаждаемый вентилятором (*cooler*). Конструктивно процессор состоит из ячеек, в которых данные могут не только храниться, но и изменяться.

С остальными устройствами компьютера, в первую очередь с оперативной памятью, процессор связан несколькими группами проводников, называемых шинами. Основных шин три: шина данных, адресная шина и командная шина (см. рис. 3.3).

*Адресная шина.* У процессоров Intel Pentium (а именно они наиболее распространены в персональных компьютерах) адресная шина 32-разрядная, т. е. состоит из 32 параллельных линий. Комбинация из 32 нулей и единиц образует 32-разрядный адрес, указывающий на одну из ячеек оперативной памяти. К ней и подключается процессор для копирования данных из ячейки в один из своих регистров.

*Шина данных.* По этой шине происходит копирование данных из оперативной памяти в регистры процессора и обратно. В компьютерах, собранных на базе процессоров Intel Pentium, шина данных 64-разрядная, то есть состоит из 64 линий, по которым за один раз на обработку поступают сразу 8 байтов.

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

Каждой шине соответствует свое адресное пространство, т. е. максимальный объем адресуемой памяти:  $2^{16} = 64$  Кб ;  $2^{20} = 1$  Мб;  $2^{24} = 16$  Мб;  $2^{32} = 4$  Гб.

В персональных компьютерах величина адресного пространства процессора и величина фактически установленной оперативной памяти практически всегда различаются. В современных персональных компьютерах с 32-разрядной шиной адреса величина адресуемой па-

мяти составляет 4 Гб, а величина фактически установленной оперативной памяти значительно меньше.

Подключение отдельных модулей компьютера к магистрали на физическом уровне осуществляется с помощью контроллеров, адаптеров устройств (*видеоадаптер, контроллер жестких дисков и т. д.*), а на программном уровне обеспечивается загрузкой в оперативную память драйверов устройств, которые обычно входят в состав операционной системы.

### *Основные параметры процессоров*

Основными параметрами процессоров являются *рабочее напряжение, разрядность, рабочая тактовая частота, коэффициент внутреннего умножения тактовой частоты и размер кэш-памяти.*

Рабочее напряжение процессора обеспечивает материнская плата, поэтому разным маркам процессоров соответствуют разные материнские платы (их надо выбирать совместно). По мере развития процессорной техники происходит постепенное понижение рабочего напряжения. Ранние модели процессоров имели рабочее напряжение 5В, а в настоящее время оно составляет менее 3В. Понижение рабочего напряжения позволяет уменьшить расстояния между структурными элементами в кристалле процессора до десятитысячных долей миллиметра, не опасаясь электрического пробоя. Пропорционально квадрату напряжения уменьшается и тепловыделение в процессоре, а это позволяет увеличивать его производительность без угрозы перегрева.

Разрядность процессора показывает, сколько бит данных он может принять и обработать в своих регистрах за один раз (за один такт). Первые процессоры были 4-разрядными. Современные процессоры семейства Intel Pentium являются 32-разрядными, хотя и работают с 64-разрядной шиной данных (разрядность процессора определяется не разрядностью шины данных, а разрядностью командной шины).

В основе работы процессора лежит тот же тактовый принцип, что и в обычных часах. Исполнение каждой команды занимает определенное количество тактов. В персональном компьютере тактовые импульсы задает одна из микросхем, входящая в микропроцессорный комплект (чипсет), расположенный на материнской плате. Чем выше частота тактов, поступающих на процессор, тем больше команд он может исполнить в единицу времени, тем выше производительность

процессора. Первые процессоры могли работать с частотой не выше 4,77 МГц, а сегодня рабочие частоты некоторых процессоров уже превосходят 3 ГГц.

Тактовые сигналы процессор получает от материнской платы, которая в отличие от процессора представляет собой не кристалл кремния, а большой набор проводников и микросхем. По физическим причинам материнская плата не может работать с такими высокими частотами, как процессор. Сегодня ее предел составляет 100–133 МГц. Для получения более высоких частот в процессоре происходит внутреннее умножение частоты на коэффициент 3; 3,5; 4; 4,5; 5 и более.

### *История и развития процессоров*

Первый микропроцессор Intel 4004 был представлен 15 ноября 1971 года корпорацией Intel. Он был 4-разрядный, содержал 2300 транзисторов, работал на тактовой частоте 108 кГц и стоил 300\$. Его сменили 8-разрядный Intel 8080 и 16-разрядный 8086, заложившие основы архитектуры всех современных процессоров.

Если бы со времени появления процессора i4004 исследователи и разработчики бездействовали и не уменьшали бы размер транзисторов, то сегодня, например, устаревший процессор Intel Core 2 Duo занимал бы весьма внушительную комнату площадью несколько десятков квадратных метров, а так кристалл процессора не превышает площадь двухрублевой монеты!

В 1965 году Гордон Мур, один из основателей корпорации Intel, лидирующей в области компьютерных интегральных схем – «чипов», высказал предположение, что число транзисторов в них будет ежегодно удваиваться. В течение последующих 10 лет это предсказание сбылось, и тогда он предположил, что теперь это число будет удваиваться каждые 2 года. Действительно, число транзисторов в микропроцессорах удваивается каждые 18 месяцев. Теперь специалисты по компьютерной технике называют эту тенденцию законом Мура.

В соответствии с законом Гордона Мура реализуется динамика изменения плотности транзисторов на кристалле процессора (рис. 4.4). В 2003 году он оценил количество всех производимых в мире транзисторов в миллиард миллиардов штук ежегодно. Это примерно в 100–150 раз больше, чем существует муравьев на Земле.



Рисунок 4.4 – Закон Мура, иллюстрирующий динамику изменения плотности транзисторов на кристалле процессора

### Процессоры Intel Core 12-го поколения

Наиболее популярные процессоры сегодня производят фирмы Intel и AMD. Лидером по продажам со значительным отрывом очень долгое время был Intel, так как раньше он имел более современные технологии производства и высокую производительность топовых моделей, но AMD с выходом процессоров Ryzen сильно укрепила свои позиции и несколько лет доминировала практически во всех ценовых категориях, и только с выходом процессоров Intel 10-го и более поздних поколений ситуация постепенно стала меняться на противоположную. В настоящее время относительный паритет по соотношению производительности и цены между процессорами AMD и Intel наблюдается только среди мощных и очень мощных процессоров, во всех остальных нишах почти полностью доминирует Intel, у которого процессоры 10–12 поколений очень выгодны для покупки<sup>1</sup>.

Распространенное мнение о том, что чем больше ядер и выше частота у процессора, тем быстрее работает компьютер, в значительной мере является неточным, так как скорость работы системы часто ощущается не по максимальной ее производительности, а по отзывчивости, т. е. скорости выполнения небольших операций типа запуска браузера или текстового редактора, которая напрямую зависит от типа жесткого диска – быстрый SSD или значительно более медленный классический HDD.

<sup>1</sup>По данным сайта <https://www.dxdigitals.info/2014/02/vibor-processora-vibirayem-samiy-luchshiy-processor.html> [37].

Самое главное в процессоре не количество ядер, а его полная производительность, но все же упрощенно можно сказать, что:

- 2 ядра – исключительно для офисной работы, Интернета и простейших игр;
- 4 ядра – для универсальных и недорогих игровых компьютеров;
- 6–8 ядер – для средних и мощных игровых компьютеров;
- 10–12 ядер – для очень мощных игровых компьютеров;
- выше 12 ядер – для экстремально мощных игровых и специализированных компьютеров (обработка фото или видео и т. п.).

Если проанализировать результаты тестов производительности процессоров и актуальные цены, то можно сделать вывод, что в первой половине 2022 года самое высокое сочетание производительности и цены (цена-качество) имеют процессоры Intel 10–12 поколений, при этом среди AMD можно назвать всего лишь несколько выгодных для покупки моделей, а вот процессоры Intel 6–9 поколений чаще всего вообще нет смысла покупать.

На рисунке 4.5 приведено изображение микропроцессора Intel Core 12-го поколения



*Рисунок 4.5 – Микропроцессор Intel Core 12-го поколения*

Представленные в 4-м квартале 2021 года процессоры Intel Core 12-го поколения Alder Lake – это шестерка топовых моделей Core i5, Core i7, Core i9 с возможностью разгона<sup>2</sup>.

---

<sup>2</sup>По данным сайта <https://remontcompa.ru/zhelezo-i-periferiya/2604-vyshli-processory-intel-core-12-go-pokolenija-alder-lake.html> [35].

### Основные новшества процессоров:

- новый принцип устройства с разными типами ядер, который предусматривает высокопроизводительные и энергоэффективные ядра, работающие в связке с технологией распределения нагрузок;
- улучшенная производительность за счет увеличения числа ядер, числа потоков и кэш-памяти;
- поддержка оперативной памяти DDR5 наряду с DDR4;
- поддержка PCI-E 5.0 наряду с PCI-E 4.0;
- новый сокет LGA 1700.

Ключевые характеристики представленных процессоров Intel Core 12-го поколения выглядят так (табл. 4.2):

Таблица 4.2 – Характеристики процессоров Intel Core 12-го поколения

Процессор Intel Core	i5-12600K	i5-12600KF	i7-12700K	i7-12700KF	i9-12900K	i9-12900KF
Общее число ядер	10	10	12	12	16	16
Число производительных ядер	6	6	8	8	8	8
Число энергоэффективных ядер	4	4	4	4	8	8
Число потоков	16	16	20	20	24	24
Максимальная частота в режиме Turbo	4.9 ГГц	4.9 ГГц	5 ГГц	5 ГГц	5.2 ГГц	5.2 ГГц
Частота производительных ядер базовая и максимальная Turbo	3.7–4.9 ГГц	3.7–4.9 ГГц	3.6–4.9 ГГц	3.6–4.9 ГГц	3.2–5.1 ГГц	3.2–5.1 ГГц
Частота энергоэффективных ядер базовая и максимальная Turbo	2.8–3.6 ГГц	2.8–3.6 ГГц	2.7–3.8 ГГц	2.7–3.8 ГГц	2.4–3.9 ГГц	2.4–3.9 ГГц
Кэш-память L3	20 Мб	20 Мб	25 Мб	25 Мб	30 Мб	30 Мб
Встроенная графика	Intel UHD Graphics 770	Нет	Intel UHD Graphics 770	Нет	Intel UHD Graphics 770	Нет
Цена	\$289–299	\$264–274	\$409–419	\$384–394	\$589–599	\$564–574

### 4.1.3. Устройства, образующие внутреннюю память

В состав внутренней памяти входят *оперативная память, кэш-память и специальная память*.

Оперативная память (ОЗУ – оперативное запоминающее устройство). Существует два типа оперативной памяти – память с произ-



вольным доступом (RAM – Random Access Memory) и память, доступная только на чтение (ROM – Read Only Memory). Процессор ЭВМ может обмениваться данными с оперативной памятью с очень высокой скоростью, на несколько порядков превышающей скорость доступа к другим носителям информации, например дискам. Внешний вид оперативной памяти приведен на рисунке 4.6.



*Рисунок 4.6 – Внешний вид оперативной памяти*

**Оперативная память с произвольным доступом (RAM)** служит для размещения программ, данных и промежуточных результатов вычислений в процессе работы компьютера. Данные могут выбираться из памяти в произвольном порядке, а не строго последовательно, как это имеет место, например, при работе с магнитной лентой.

**Память, доступная только на чтение (ROM)** используется для постоянного размещения определенных программ. В процессе работы компьютера содержимое этой памяти не может быть изменено.

Оперативная память – энергозависимая, т. е. данные в ней хранятся только до выключения ПК. Для долговременного хранения информации служат дискеты, винчестеры, компакт-диски и т. п.

Конструктивно элементы памяти выполнены в виде модулей, так что при желании можно сравнительно просто заменить их или установить дополнительные и тем самым изменить объем общей оперативной памяти компьютера. Емкость модулей памяти кратна степени числа 2: 128, 256, 512, 1024 Мб...

#### *Виды RAM*

**Полупроводниковая статическая (SRAM)** – ячейки представляют собой полупроводниковые триггеры. Достоинства – небольшое энергопотребление, высокое быстродействие. Недостатки – малый объем, высокая стоимость. Сейчас широко используется в качестве кеш-памяти процессоров.

**Полупроводниковая динамическая (DRAM)** – каждая ячейка представляет собой конденсатор. Достоинства – низкая стоимость,



большой объем. Недостатки – необходимость периодического считывания и перезаписи каждой ячейки – так называемой регенерации, и как следствие, понижение быстродействия, большое энергопотребление. Обычно используется в качестве оперативной памяти компьютеров.

Модули памяти характеризуются такими параметрами, как *объем* (Мбайт), *число микросхем*, *паспортная частота* (МГц), *время доступа к данным* (наносекунды) и *число контактов*.

*Кэш-память*. Обмен данными внутри процессора происходит в несколько раз быстрее, чем обмен с другими устройствами, например, с оперативной памятью. Для того чтобы уменьшить количество обращений к оперативной памяти, внутри процессора создают буферную область – так называемую кэш-память (англ. *cache*). Это своего рода сверхоперативная память. Когда процессору нужны данные, он сначала обращается в кэш-память, и только если там нужных данных нет, происходит его обращение в оперативную память. При этом возможны как попадания, так и промахи. В случае попадания, т. е. если в кэш подкачаны нужные данные, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше отсутствует, то процессор считывает ее непосредственно из оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования. Высокопроизводительные процессоры комплектуют повышенным объемом кэш-памяти.

Кэш-памятью управляет специальное устройство – контроллер, который, анализируя выполняемую программу, пытается предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш-память.

Нередко кэш-память распределяют по нескольким уровням. Кэш первого уровня выполняется в том же кристалле, что и сам процессор, и имеет объем порядка десятков Кбайт.

Кэш второго уровня находится либо в кристалле процессора, либо в том же узле, что и процессор, хотя и выполняется на отдельном кристалле. Кэш-память первого и второго уровня работает на частоте, согласованной с частотой ядра процессора.

Кэш-память третьего уровня выполняют на быстродействующих микросхемах типа *SRAM* и размещают на материнской плате вблизи процессора. Ее объемы могут достигать нескольких Мбайт, но работает она на частоте материнской платы.

#### *Специальная память*

К устройствам специальной памяти относят постоянную память (*ROM*), перепрограммируемую постоянную память (*Flash Memory*),

память *CMOS RAM*, питаемую от батарейки, видеопамять и некоторые другие виды памяти.

Постоянная память (ПЗУ, англ. ROM, *Read Only Memory* – память только для чтения) – энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения. Содержание памяти специальным образом зашивается в устройстве при его изготовлении для постоянного хранения. Из ПЗУ можно только читать. Прежде всего, в постоянную память записывают программу управления работой самого процессора. В ПЗУ находятся программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Перепрограммируемая постоянная память (*Flash Memory*) – энергонезависимая память, допускающая многократную перезапись своего содержимого с дискеты.

Важнейшая микросхема постоянной памяти – модуль BIOS. Роль BIOS двоякая: с одной стороны это неотъемлемый элемент аппаратуры, а с другой стороны – важный модуль любой операционной системы.

BIOS (*Basic Input/Output System* – базовая система ввода-вывода) – совокупность программ, предназначенных для автоматического тестирования устройств после включения питания компьютера и загрузки операционной системы в оперативную память (рис. 4.5).

Разновидность постоянного ЗУ – CMOS RAM.

CMOS RAM – это память с невысоким быстродействием и минимальным энергопотреблением от батарейки. Используется для хранения информации о конфигурации и составе оборудования компьютера, а также о режимах его работы (рис. 4.7).



Рисунок 4.7 – Интегральные схемы BIOS и CMOS

Содержимое CMOS изменяется специальной программой Setup, находящейся в BIOS (англ. *Set-up* – устанавливать).

#### ***4.1.4. Жесткий диск, или винчестер***

Накопитель на жестких магнитных дисках (НЖМД), жесткий диск, или винчестер (англ. *Hard Disk Drive, HDD*) – *энергонезависимое*, перезаписываемое компьютерное запоминающее устройство. Является основным накопителем данных практически во всех современных компьютерах.

Информация в НЖМД записывается на жесткие (алюминиевые или стеклянные) пластины, покрытые слоем ферромагнитного материала (платтеры). Считывающие головки в рабочем режиме не касаются поверхности пластин благодаря прослойке воздуха, образуемой при быстром вращении дисков. Головки считывания-записи вместе с их несущей конструкцией и дисками заключены в герметически закрытый корпус, называемый модулем данных. При установке модуля данных на дисковод он автоматически соединяется с системой, подкачивающей очищенный охлажденный воздух. Поверхность платтера имеет магнитное покрытие толщиной всего лишь в 1,1 мкм, а также слой смазки для предохранения головки от повреждения при опускании и подъеме на ходу. При вращении платтера над ним образуется воздушный слой, который обеспечивает воздушную подушку для зависания головки на высоте 0,5 мкм над поверхностью диска (рис. 4.8).



*Рисунок 4.8 – Винчестерский накопитель со снятой крышкой корпуса*

Способ записи двоичной информации на магнитной среде называется магнитным кодированием. Он заключается в том, что магнит-

ные домены в среде выстраиваются вдоль дорожек в направлении приложенного магнитного поля своими северными и южными полюсами. Обычно устанавливается однозначное соответствие между двоичной информацией и ориентацией магнитных доменов.

Информация записывается по концентрическим дорожкам (трекам), которые делятся на секторы (рис. 4.9). Количество дорожек и секторов зависит от формата винчестера. Сектор хранит минимальную порцию информации, которая может быть записана на диск или считана. Емкость сектора постоянна и составляет 512 байтов.

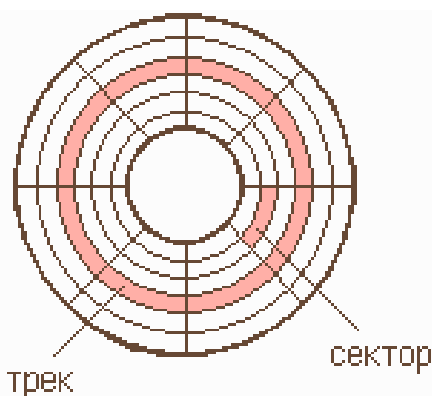


Рисунок 4.9 – Поверхность магнитного диска

Название «винчестер» жесткий диск получил благодаря фирме IBM, которая в 1973 выпустила жесткий диск модели 3340, впервые объединивший в одном неразъемном корпусе диски и считывающие головки. При его разработке инженеры использовали краткое внутреннее название «30-30», что означало два модуля (в максимальной компоновке) по 30 Мб каждый. Кеннет Хотон, руководитель проекта, по созвучию с обозначением популярного охотничьего ружья «Winchester 30-30» предложил назвать этот диск «винчестером».

В Европе и Америке название «винчестер» вышло из употребления в 1990-х годах; в российском же компьютерном сленге название «винчестер» сохранилось, сократившись до слова «винт».

#### *Характеристики НЖМД*

Интерфейс – способ, использующийся для передачи данных. Современные накопители могут использовать интерфейсы SCSI, SAS, SATA, FireWire, USB и Fibre Channel.

Емкость – количество данных, которые могут храниться накопителем. Емкость современных устройств может достигать до 1.5 Тб и более. В отличие от принятой в информатике системе приставок, обо-

значающих кратную 1024 величину (кило=1024), производителями при обозначении емкости жестких дисков используются кратные 1000 величины. Например, настоящая емкость жесткого диска, маркированного как «200 Гб», составляет 186,2 Гб.

Физический размер – почти все современные накопители для персональных компьютеров и серверов имеют размер либо 3,5, либо 2,5 дюйма. Последние, чаще применяются в ноутбуках. Появилась информация о создании устройств с размерами менее 2 дюймов.

Скорость вращения шпинделя – количество оборотов шпинделя в минуту. От этого параметра в значительной степени зависят время доступа и скорость передачи данных. В настоящее время выпускаются винчестеры со следующими стандартными скоростями вращения: 4200, 5400 и 7200 (ноутбуки), 7200 и 10000 (персональные компьютеры), 10000 и 15000 об./мин. (серверы и высокопроизводительные рабочие станции).

#### **4.1.5. Графическая плата**

Графическая плата (известна также как графическая карта, видеокарта, видеоадаптер) (англ. *videocard*) – устройство, преобразующее изображение, находящееся в памяти компьютера, в видеосигнал для монитора.

Первый IBM PC не предусматривал возможности вывода графических изображений. Современный ПК позволяет выводить на экран двух- и трехмерную графику и полноцветное видео.

Обычно видеокарта является платой расширения и вставляется в специальный разъем (ISA, VLB, PCI, AGP, PCI-Express) для видеокарт на материнской плате, но бывает и встроенной (рис. 4.10).



Рисунок 4.10 – Внешний вид видеокарты

Современная графическая плата состоит из следующих основных частей:

- графический процессор (GPU) занимается расчетами выводимого изображения, освобождая от этой обязанности центральный процессор, производит расчеты для обработки команд трехмерной графики. Является основой графической платы, именно от него зависят быстродействие и возможности всего устройства. Современные графические процессоры по сложности мало чем уступают центральному процессору;

- видеоконтроллер отвечает за формирование изображения в видеопамяти;

- видеопамять выполняет роль буфера, в котором в цифровом формате хранится изображение, предназначенное для вывода на экран монитора. Емкость видеопамяти так же, как и оперативной памяти, кратна степени числа два и на сегодняшний день измеряется в мегабайтах;

- цифро-аналоговый преобразователь (ЦАП) служит для преобразования изображения, формируемого видеоконтроллером, в уровни интенсивности цвета, подаваемые на аналоговый монитор. Большинство ЦАП имеют разрядность 8 бит на канал – получается по 256 уровней яркости на каждый основной цвет RGB, что в сумме дает 16.7 млн цветов.

#### ***4.1.6. Звуковая плата***

Звуковую плату (также называемая звуковая карта, аудиоадаптер) используют для записи и воспроизведения различных звуковых сигналов: речи, музыки, шумовых эффектов.

IBM-PC проектировался не как мультимедийная машина, а инструмент для решения серьезных научных и деловых задач, звуковая карта на нем не была предусмотрена и даже не запланирована. Единственный звук, который издавал компьютер, был звук встроенного динамика бипера, сообщавший о неисправностях.

Любая современная звуковая карта может использовать несколько способов воспроизведения звука. Одним из простейших является преобразование ранее оцифрованного сигнала снова в аналоговый. Глубина оцифровки сигнала (например, 8 или 16 бит) определяет качество записи и, соответственно, воспроизведения. Так,



8-разрядное преобразование обеспечивает качество звучания кассетного магнитофона, а 16-разрядное – качество компакт-диска.

В настоящее время звуковые карты чаще бывают встроенными в материнскую плату, но выпускаются также и как отдельные платы расширения (рис. 4.11).



*Рисунок 4.11 – Звуковая плата*

На материнскую плату звуковая плата устанавливается в слоты ISA (устаревший формат) или PCI (современный формат). Когда звуковая плата установлена, на задней панели корпуса компьютера появляются порты для подключения колонок, наушников, микрофона.

#### **4.1.7. Сетевая плата**

Сетевая плата (также известная как сетевая карта, сетевой адаптер, *Ethernet card*, *NIC* (англ. *network interface card*)) – печатная плата, позволяющая взаимодействовать компьютерам между собой, посредством локальной сети.

Обычно, сетевая плата идет как отдельное устройство и вставляется в слоты расширения материнской платы (в основном – PCI, ранние модели использовали шину ISA). На современных материнских платах сетевой адаптер все чаще является встроенным, таким образом, покупать отдельную плату не нужно (рис. 4.12).

На сетевой плате имеются разъемы для подключения кабеля витой пары.

Сетевая карта относится к устройствам коммуникации (связи). Скорость передачи данных устройствами коммуникации измеряется в битах в секунду (а также в Мбит/с и Гбит/с).

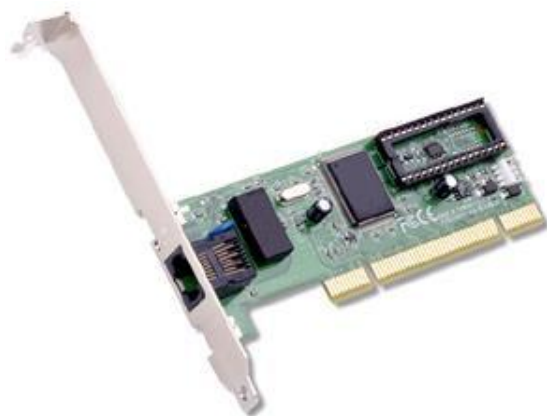


Рисунок 4.12 – Сетевая карта

*Накопители на DVD дисках.* DVD (Digital Versatile Disc, цифровой многоцелевой, или универсальный, диск) – это оптические диски большой емкости, которые применяют для хранения полнометражных фильмов, музыки высокого качества, компьютерных программ.

Существует несколько вариантов DVD, отличающихся по емкости: односторонние и двухсторонние, однослойные и двухслойные.

Односторонние однослойные DVD имеют емкость 4,7 Гбайт информации, двухслойные – 8,5 Гбайт; двухсторонние однослойные вмещают 9,4 Гбайт, двухслойные – 17 Гбайт.

Луч лазера в устройствах DVD имеет длину волны от 635 нм до 650 нм, благодаря чему плотность записи DVD существенно выше (рис. 4.13).

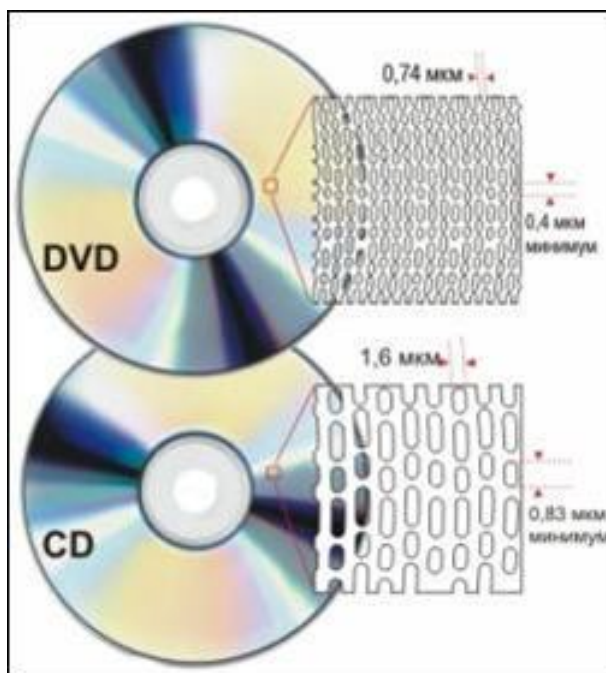


Рисунок 4.13 – Сравнение плотности записи на CD и DVD



Разработчики DVD ориентировались прежде всего на возможность записи целого видеофильма с качеством MPEG-2 на один диск, поэтому средняя скорость считывания видеоинформации составляет 4,692 Мбит/с (примерно 600 Кбайт/с), из которых собственно видео считывается со скоростью 3,5 Мбит/с, аудиопоток на трех языках в шестиканальном стандарте Dolby Surround – со скоростью 1,16 Мбит/с, а субтитры на 4 языках (из 32 возможных) – со скоростью 40 Кбит/с. Эта скорость в DVD принята за однократную (1x). Умножив скорость 1x потока на стандартную продолжительность фильма (133 минуты), получаем минимальный объем DVD – 4,7 Гбайт.

Помимо чтения данных с DVD со скоростью порядка 1,2 Мбайт/с, накопители DVD способны читать обычные CD-ROM со скоростью, примерно соответствующей 8–10-скоростным приводам CD-ROM.

В настоящее время массово эксплуатируются устройства DVD, позволяющие записывать и перезаписывать данные.

**Флэш-память.** Флэш-память (*flash*) – разновидность полупроводниковой энергонезависимой перезаписываемой памяти.

Флэш-память может быть прочитана сколько угодно раз, но писать в такую память можно лишь ограниченное число раз (обычно около 10 тысяч). Причина в том, что для записи в память необходимо сначала стереть участок памяти, а участок может выдержать лишь ограниченное число стираний.

Преимуществом флэш-памяти над оперативной является ее энергонезависимость – при выключении энергии содержимое памяти сохраняется.

Преимуществом флэш-памяти над жесткими дисками, CD и DVD дисками является отсутствие движущихся частей. Поэтому флэш-память более компактна, дешева (с учетом стоимости устройств чтения-записи) и обеспечивает более быстрый доступ.

Объем современных флэш-карт составляет 64, 128, 256 и более Гб (рис. 4.14.).



Рисунок 4.14 – Устройства на основе flash-памяти

Благодаря своей компактности, дешевизне и отсутствию потребности в энергии, флэш-память широко используется в портативных устройствах, работающих на батарейках и аккумуляторах – цифровых фотокамерах и видеокамерах, цифровых диктофонах, MP3-плеерах и т. д.

## 4.2. Периферийные внешние устройства

Периферийными называются устройства внешние по отношению к системному блоку. Обычно они служат для ввода/вывода информации при взаимодействии человек-компьютер.

К основным устройствам *ввода* относятся *клавиатура, мышь, сканер*, к основным устройствам *вывода* – *монитор, принтер*.

Клавиатура (*keyboard*) содержит 101 или 104 клавиши (рис. 4.15). Стандартом расположения символьных клавиш является раскладка QWERTY (ЙЦУКЕН) по названию клавиш верхнего символьного ряда слева направо.

Области клавиатуры

1. Алфавитно-цифровая.
2. Специальных клавиш <Alt> <Ctrl> <Shift> <Caps Lock> <Enter> <Delete> <←> <Insert> <Print Screen>...
3. Управления курсором.
4. Переключаемая (цифровая / управления курсором). Режимы переключаются клавишей <Num Lock>.
5. Функциональная <F1> – <F12>.
6. Индикаторов.

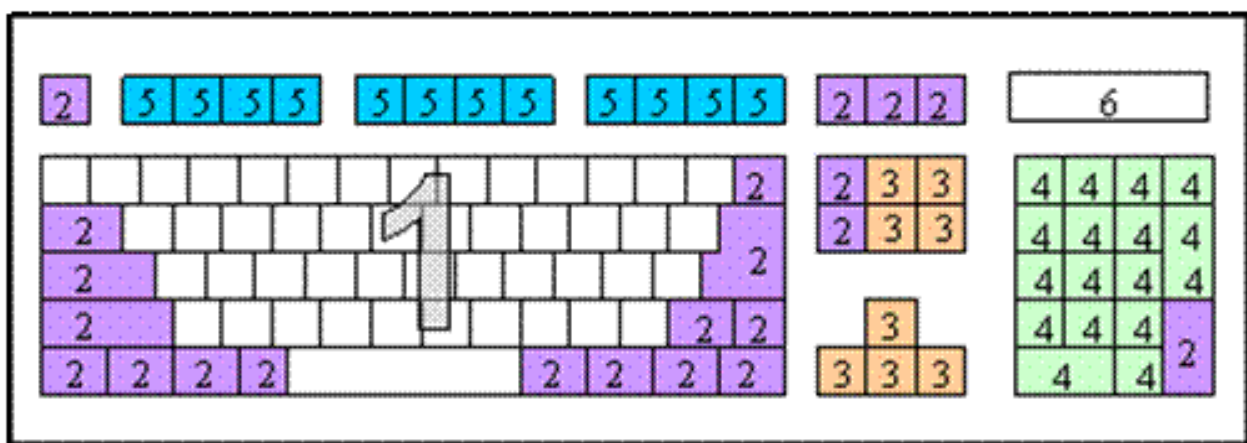


Рисунок 4.15 – Разделение клавиатуры на области

Предназначение некоторых специальных клавиш:

1. Esc – отмена, отказ.
2. Tab – табулирование.
3. Del – удаление символа справа от курсора.
4. ← – «забой», удаление символа слева от курсора.
5. Ins – клавиша переключения режима вставки / замены символов.
6. Home – перевод курсора в начало строки.
7. End – перевод курсора в конец строки.
8. PgUp – переход на страницу вверх.
9. PgDn – переход на страницу вниз.
10. Enter – клавиша ввода.
11. Break – прерывание.
12. Shift – смена верхнего / нижнего регистра при удержании.
13. Caps Lock – смена верхнего / нижнего регистра.
14. Print Screen – копирование текущего состояния экрана монитора в буфер обмена.

Манипуляторы, или координатные устройства ввода информации, являются неотъемлемой частью современного компьютера. Наиболее известны следующие типы манипуляторов: мышь, трекбол, графические планшеты, устройства ввода, применяемые в ноутбуках – тачпад (сенсорная панель), а также джойстики.

*Компьютерная мышь.* Первую компьютерную мышь создал Дуглас Энджелбарт в 1963 году в Стэнфордском исследовательском центре. Распространение мыши получили благодаря росту популярности программных систем с графическим интерфейсом пользователя. Мышь делает удобным манипулирование такими широко распространенными в графических пакетах объектами, как окна, меню, кнопки, пиктограммы и т. д.

#### *Подключение мыши к компьютеру*

Изначально для подключения мыши к компьютеру использовался провод (в обиходной речи «хвост») который подключался в один из портов компьютера. Первым из широко применяемых стандартных портов стал СОМ-порт, впоследствии его сменил порт PS/2, который в настоящее время вытесняется портом USB (рис. 4.16).

Провод часто являлся помехой при работе с мышью, поэтому от него неоднократно пытались избавиться. Первыми попытками было внедрение инфракрасной связи между мышью и специальным приемным устройством, которое, в свою очередь, подключалось к порту компьютера. Но оптическая связь, как показала практика, тоже не

лишена недостатка, любое препятствие между мышью и датчиком мешало работе.



*Рисунок 4.16 – Различные варианты компьютерных мышей*

Трекбол представляет собой «перевернутую» мышь: в движение приводится не сам корпус устройства, а только его шар. Это позволяет существенно повысить точность управления курсором и, кроме того, экономить место, поэтому трекболы часто используют в ноутбуках (рис. 4.17).



*Рисунок 4.17 – Оптический трекбол*

Графический планшет (или дигитайзер, диджитайзер). Графический планшет это устройство для ввода рисунков от руки непосредственно в компьютер. Состоит из пера и плоского планшета, чувствительного к нажатию пера. Также к планшету может прилагаться специальная мышь (рис. 4.18).



Рисунок 4.18 – Графический планшет

Тачпад (сенсорная панель) (*touchpad* или *trackpad*) – это устройство ввода, применяемое в ноутбуках, служит для перемещения курсора в зависимости от движений пальца пользователя (рис. 4.19). Используется в качестве замены компьютерной мыши.



Рисунок 4.19 – Тачпад (сенсорная панель)

Сенсорные панели различаются по размерам, но обычно их площадь не превосходит  $50 \text{ см}^2$ . Работа сенсорной панели основана на измерении емкости пальца или измерении емкости между сенсорами. Емкостные сенсоры расположены вдоль вертикальной и горизонтальной осей панели, что позволяет определять положение пальца с нужной точностью. Поскольку работа устройства основана на измерении емкости, оно не будет работать, если водить по нему каким-либо непроводящим предметом, например, основанием карандаша. В случае использования проводящих предметов сенсорная панель будет работать только при достаточной площади соприкосновения, поэтому, например, работа с влажными пальцами весьма затруднена. Преимуществами сенсорных панелей являются:

- отсутствие необходимости в ровной поверхности, как для мыши;
- расположение сенсорной панели, как правило, фиксировано относительно клавиатуры;
- для перемещения курсора на весь экран достаточно лишь небольшого перемещения пальца;
- работа с ними не требует особого привыкания, как, например, в случае с трекболом.

Недостатком сенсорных панелей является низкое разрешение, что затрудняет работу в графических редакторах и 3D-играх.

Джойстик является аналоговым координатным устройством ввода информации, выполняемым обычно в виде двух реостатных датчиков с питанием +5 В (рис. 4.20). Рукоятка джойстика связана с двумя переменными резисторами, изменяющими свое сопротивление при ее перемещении. Один резистор определяет перемещение по координате X, другой – по Y. Джойстик обычно подключается к адаптеру игрового порта, расположенному на многофункциональной плате ввода-вывода (Multi I/O Card) или звуковой карте (в последнем случае разъем игрового порта совмещается с интерфейсом MIDI).



*Рисунок 4.20 – Джойстик*

Сканер (англ. *scanner*) – устройство, которое создает цифровое изображение сканируемого объекта. Полученное изображение может быть сохранено как графический файл, или, если оригинал содержал текст, распознано посредством программы распознавания текста и сохранено как текстовый файл.

Рассмотрим принцип действия планшетных сканеров, как наиболее распространенных моделей (рис. 4.21). Сканируемый объект



кладется на стекло планшета сканируемой поверхностью вниз. Под стеклом располагается подвижная лампа, движение которой регулируется шаговым двигателем.

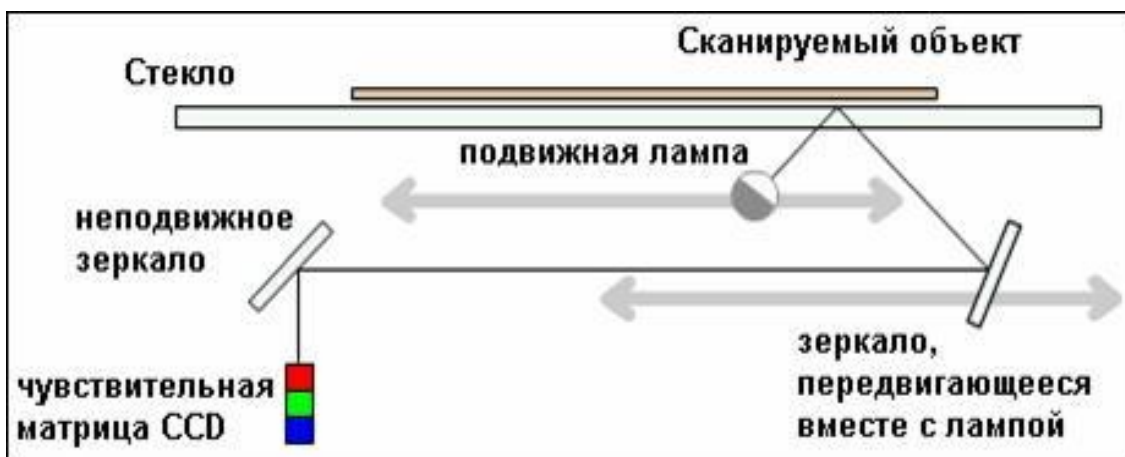


Рисунок 4.21 – Устройство планшетного сканера

Свет, отраженный от объекта, через систему зеркал попадает на чувствительную матрицу (CCD – Couple-Charged Device), далее на АЦП и передается в компьютер. За каждый шаг двигателя сканируется полоска объекта, потом все полоски объединяются программным обеспечением в общее изображение.

В зависимости от способа сканирования объекта и самих объектов сканирования существуют следующие виды сканеров:

Планшетные – наиболее распространенные, поскольку обеспечивают максимальное удобство для пользователя – высокое качество и приемлемую скорость сканирования. Представляет собой планшет, внутри которого под прозрачным стеклом расположен механизм сканирования (рис. 4.22).



Рисунок 4.22 – Планшетный сканер

Ручные – в них отсутствует двигатель, следовательно, объект приходится сканировать вручную, единственным его плюсом является дешевизна и мобильность, при этом он имеет массу недостатков: низкое разрешение, малую скорость работы, узкая полоса сканирования, возможны перекосы изображения, поскольку пользователю будет трудно перемещать сканер с постоянной скоростью (рис. 4.23).



*Рисунок 4.23 – Ручной сканер*

Листопротяжные – лист бумаги вставляется в щель и протягивается по направляющим роликам внутри сканера мимо лампы. Имеет меньшие размеры по сравнению с планшетным сканером, однако может сканировать только отдельные листы. Многие модели имеют устройство автоматической подачи, что позволяет быстро сканировать большое количество документов, причем в ряде моделей – с двух сторон за один прогон (рис. 4.24).



*Листопротяжный сканер*

*Рисунок 4.24 – Лентопротяжный сканер*



Планетарные – применяются для сканирования книг или легко повреждающихся документов. При сканировании нет контакта со сканируемым объектом как в планшетных сканерах (рис. 4.25).



*Рисунок 4.25 – Планетарный сканер*

Барабанные – применяются в полиграфии, имеют большое разрешение (около 10 тысяч точек на дюйм). Оригинал располагается на внутренней или внешней стенке прозрачного цилиндра – барабана (рис. 4.26).



*Рисунок 4.26 – Барабанный сканер*

Слайд-сканеры – как ясно из названия, служат для сканирования пленочных слайдов, выпускаются как самостоятельные устройства, так и в виде дополнительных модулей к обычным сканерам (рис. 4.27).



*Рисунок 4.27 – Слайд-сканер*

Сканеры штрих-кода – небольшие, компактные модели для сканирования штрих-кодов товара в магазинах (рис. 4.28).



Рисунок 4.28 – Сканеры штрих-кода

Одним из основных устройств *вывода* информации является монитор.

Монитор – устройство визуального представления данных. Это не единственно возможное, но главное устройство вывода. Его основными потребительскими параметрами являются: тип, размер и шаг маски экрана, максимальная частота регенерации изображения, класс защиты. *Виды мониторов:* мониторы электронно-лучевые (CRT) (в настоящее время не используются), мониторы жидкокристаллические (LCD), плазменные панели.

Сейчас наиболее распространены плоские жидкокристаллические мониторы (ЖК). Они пришли на смену мониторам, основанным на электронно-лучевой трубке (ЭЛТ).

*Размер монитора* измеряется между противоположными углами видимой части экрана по диагонали. Единица измерения – дюймы. Стандартные размеры: 14»; 15»; 17»; 19»; 20»; 21», 24». В настоящее время наиболее универсальными являются мониторы размером 21 и 24 дюймов.

*Разрешающая способность* монитора характеризуется числом точек выводимого изображения. Принято указывать отдельно количество точек по горизонтали и вертикали. Например, разрешение монитора 1024x768 означает возможность различить до 1024 точек по горизонтали при числе строк до 768.

На экране жидкокристаллического монитора изображение образуется в результате прохождения белого света лампы подсветки через ячейки, прозрачность которых зависит от приложенного напряжения. Элементарная триада состоит из трех ячеек зеленого, красного и си-

него цвета и соответствует одному пикселу экрана. Размер монитора по диагонали и разрешение экрана однозначно определяют размер такой триады и тем самым *зернистость* изображения.

*Частота регенерации* (обновления) изображения показывает, сколько раз в течение секунды монитор может полностью сменить изображение (поэтому ее также называют частотой кадров). Этот параметр зависит не только от монитора, но и от свойств и настроек видеоадаптера, хотя предельные возможности определяет все-таки монитор. Частоту регенерации изображения измеряют в герцах (Гц).

Большинством параметров изображения, полученного на экране монитора, можно управлять программно.

ЭЛТ и плазменные мониторы, как практически не производящиеся в настоящее время, рассматривать не будем.

Жидкокристаллический монитор (*liquid crystal display, LCD*) – плоский монитор на основе жидких кристаллов. *LCD TFT* (TFT – *thin film transistor* – тонкопленочный транзистор) – одно из названий жидкокристаллического дисплея, в котором используется активная матрица, управляемая тонкопленочными транзисторами. На рисунке 4.29 представлены примеры используемых LCD-мониторов.



Рисунок 4.29 – Жидкокристаллические LCD-мониторы

Жидкокристаллические мониторы являются в настоящее время наиболее распространенным типом экранов для ПК, ноутбуков и других видов вычислительных систем. Они имеют целый ряд достоинств, таких как высокое качество изображения, малую толщину и вес, низкое энергопотребление и другие.

Есть и недостатки, например, только одно разрешение, соответствующее количеству активных элементов. При других разрешениях качество изображения резко падает. Также отмечается относительно низкая скорость переключения пикселей экрана. Несмотря на это именно эта технология является сейчас основной в данной области.

Основные типы ЖК мониторов определяются физическими размерами их матриц и соответствующими разрешениями (количеством точек – пикселей). Например, широко распространены разрешения:  $1280 \times 1024$ ,  $1920 \times 1080$  – HD,  $3840 \times 2160$  4K.

Также различаются *типы матриц* – конкретные реализации технологии ЖК. Здесь можно выделить:

*TN «Twisted Nematic»* наиболее дешевые, отличаются высокой скоростью реакции пикселей. По всем остальным параметрам проигрывают более современным матрицам.

*IPS «In-Plane Switching»* – одна из самых дорогих технологий, отличается хорошей цветопередачей и широкими углами обзора. К основным недостаткам можно отнести низкую реакцию пикселей.

*VA «Multidomain Vertical Alignment»* – технология, в которой попытались объединить преимущества TN и IPS. В результате получилось нечто среднее между ними. Существует большое количество способов реализации каждой из этих технологий (например, PLS), а также постоянно предлагаются принципиально новые решения. Поэтому выбор типа матрицы для компьютера – задача нетривиальная.

Технология OLED-дисплея (рис. 4.30) принципиально отличается от существующих моделей ЖК и в большинстве случаев намного превосходит по характеристикам почти всех параметров качества изображения. В отличие от LED OLED-телевизоры вообще не имеют подсветки. Вместо этого каждый пиксель содержит органический светодиод, который загорается или полностью выключается в зависимости от прохождения электротока.

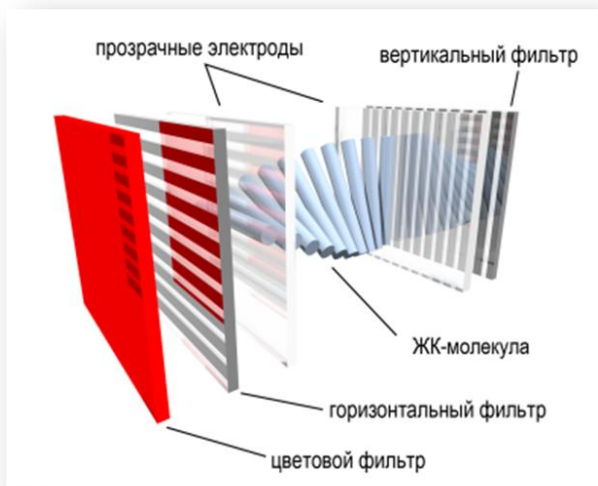


Рисунок 4.30 – Технология OLED-дисплея

В результате OLED способны обеспечить идеальное локальное затемнение с точностью до одного светодиода, а значит отличную контрастность. Цвет в OLED-телевизорах формируется голубыми и желтыми светодиодами, свет которых пропускается затем через красные или зеленые фильтры с целью создания всех цветов, в том числе и белого. В общем, формирование цвета не прямое. Также OLED-телевизоры обладают меньшей пиковой яркостью, чем лучшие LCD.

Технология QLED (рис. 4.31) – это реальная технология потенциального будущего, использующая светодиоды на основе квантовых точек. В настоящее время такие экраны практически идентичны LED-дисплеям, только с некоторыми изменениями в части улучшения цветопередачи и углов обзора. То есть QLED сейчас – LCD-экран, использующий пленку с нанесенными на нее квантовыми точками.

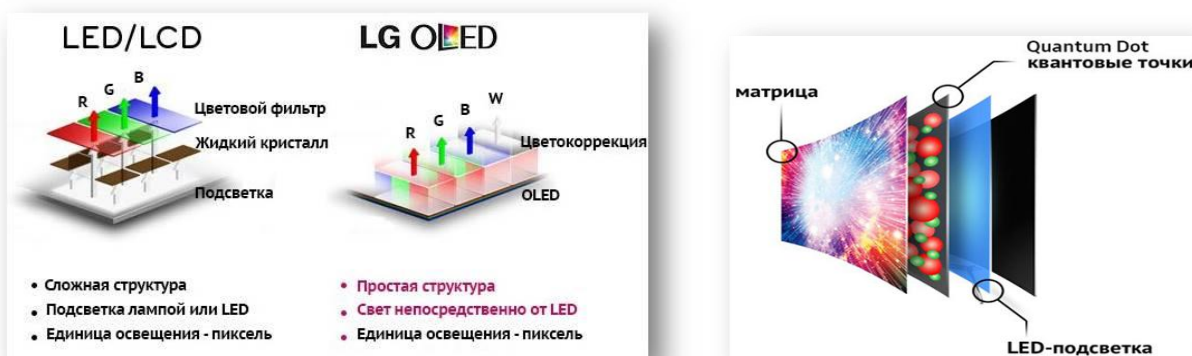


Рисунок 4.31 – Технология QLED-дисплея

Сами квантовые точки, т. е. нанокристаллы, были впервые исследованы в конце прошлого века советским физиком Алексеем Еки-



мовым, который объяснил зависимость их оптических свойств от физического размера и уровня, пропускаемого через них тока.

Главное отличие LED от QLED как раз и состоит лишь в том, что модели QLED используют новые специализированные металлические нанопоры на квантовых точках, расположенные на пленочной матрице над светодиодной панелью подсветки. В целом, это дает нам более яркие, более насыщенные цвета.

В настоящее время идут разработки более продвинутой, реальной технологии дисплеев QLED, когда пиксели будут состоять из реагирующих на свет нанокристаллов, которые при прохождении через них электрического тока не только светятся, но также окрашиваются в различные цвета.

Эта новая разработка сделает будущие QLED очень похожими на OLED-телевизоры в фундаментальном смысле. Но только цветопередача будет гораздо лучше, поскольку каждая квантовая точка имеет свой цвет, а не получает его смешением RGB цветов субпикселей, как в случае с OLED.

*Веб-камера* – устройство ввода изображения в компьютер. Они могут быть встроенными или выделенными в отдельный блок (рис. 4.32).



Рисунок 4.32 – Веб-камеры

Последние модели веб-камер отличаются многофункциональностью. Помимо прямого назначения – использования в роли веб-камеры, их можно задействовать для сканирования документов, видеопрезентаций, применять в качестве лазерной указки или диктофона.

*Принтер* (от англ. *printer* – печатник) – устройство печати информации на твердый носитель, обычно на бумагу. Процесс печати называется выводом на печать, а результат – распечаткой.

Принтеры, в зависимости от вида печати, разделяют на цветные и монохромные, в зависимости от способа нанесения изображения на матричные, струйные, лазерные.

Изображение, получаемое с помощью современных принтеров, состоит из точек (*dots*). Чем меньше эти точки и чем чаще они расположены, тем выше качество изображения. Максимальное количество точек, которые принтер может отдельно напечатать на отрезке в 1 дюйм (25,4 мм), называется разрешением и характеризуется в точках на дюйм (*dpi – dot per inch*). Хорошее качество печати обеспечивается разрешением 300 dpi и выше.

Матричные принтеры – старейший из ныне применяемых типов принтеров, его механизм был изобретен в 1964 году компанией Seiko Epson (рис. 4.33).



*Рисунок 4.33 – Один из первых образцов матричного принтера*

Изображение формируется печатной головкой, которая состоит из набора иголок, приводимых в действие электромагнитами (игольчатая матрица). Иголки ударяют по бумаге через красящую ленту, головка передвигается построчно вдоль листа.

Основными недостатками данного типа принтеров являются низкая скорость работы и высокий шум, однако благодаря дешевизне копии (расходным материалом, по сути, является только красящая лента) и возможности работы с непрерывной (рулонной, фальцованной) и копировальной бумагой они незаменимы, когда требуется печать на непрерывной бумаге (лаборатории, промышленность, бухгалтерия, ведение отчетов, печать чеков в магазинах, банкоматах и т. п.), многослойных бланках (например, авиабилеты), или минимальная стоимость печати. Сам факт ударной печати затрудняет внесение несанкционированных изменений в документ (финансовая сфера).

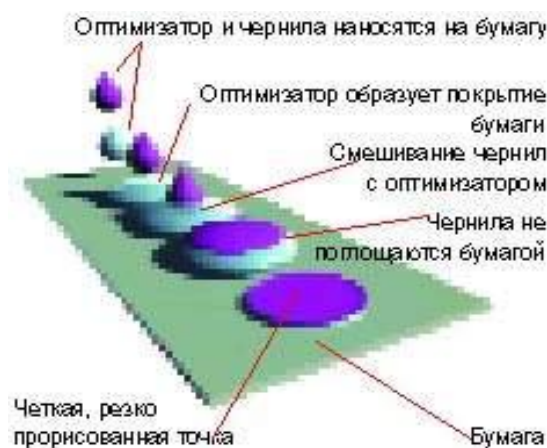
*Струйные* принтеры. Первый работающий по этой технологии принтер появился в 1976 году – это был принтер от компании IBM (рис. 4.34).



*Рисунок 4.34 – Один из первых образцов струйного принтера образца 1984 года*

Принцип печати последовательный, безударный. Изображение формируется из микрокапель (~ 50 мкм) чернил, которые выдуваются из сопел картриджа. Засорение сопел, а точнее засыхание чернил в соплах – это существенный конструктивный недостаток струйных принтеров (рис. 4.35).

Каждая строка цветного изображения проходится 4 раза (СМУК). Количество сопел обычно от 16 до 64, но есть печатающие головки с сотнями сопел.



*Рисунок 4.35 – Принцип работы струйного принтера*

Преимущества струйных принтеров:

- высокое качество графики даже для самых дешевых моделей;
- низкая стоимость принтера (продается ниже себестоимости, окупается для производителя за счет дорогих расходных материалов);
- наличие принтеров больших форматов (от А4 до А0).



Недостатки струйных принтеров:

– низкая экономичность. Затраты на чернила уже в первый год как минимум в 5 раз превысят стоимость устройства, при объемах печати в 10–15 страниц в день. Непроизводительный расход чернил на прочистку головок. Низкая емкость картриджей;

– требователен к бумаге. Для качественной печати необходима специальная бумага для струйных принтеров;

– низкая стойкость отпечатков (выцветают и смываются);

– относительно низкая надежность и скорость печати.

*Лазерные принтеры* менее требовательны к бумаге, чем, например, струйные, а стоимость печати одной страницы текстового документа у них в несколько раз ниже. Большинство представленных на рынке лазерных принтеров предназначены для черно-белой печати; цветные лазерные принтеры пока дороги и рассчитаны на корпоративных пользователей (рис. 4.36).



*Рисунок 4.36 – Один из первых образцов лазерного принтера*

Лазерные принтеры печатают на бумаге плотностью от 60 г/м<sup>3</sup> со скоростью от 8 до 24 листов в минуту (*ppm – page per minutes*), при этом разрешение может быть 1200 dpi и более. Качество текста, напечатанного на лазерном принтере с разрешением 300 dpi, примерно соответствует типографскому. Однако если страница содержит рисунки, содержащие градации серого цвета, то для получения качественного графического изображения потребуется разрешение не ниже 600 dpi. При разрешающей способности принтера 1200 dpi отпечаток получается почти фотографического качества. Если необходимо печатать большое количество документов (например, более 40 листов в день), лазерный принтер представляется единственным разумным выбором.

Технология – прародитель современной лазерной печати появилась очень давно. В 1938 году Честер Карлсон изобрел способ печати, названный электрография, а затем переименованный в ксерографию.

Сердцем лазерного принтера является фото-барабан. С его помощью производится перенос изображения на бумагу (рис. 4.37).

Фото-барабан представляет собой металлический цилиндр, покрытый тонкой пленкой фоточувствительного полупроводника. Поверхность такого цилиндра можно снабдить положительным или отрицательным электростатическим зарядом, который сохраняется до тех пор, пока барабан не освещен. Если какую-либо часть барабана осветить, покрытие приобретает проводимость и заряд стекает с освещенного участка, образуя незаряженную зону. Это ключевой момент в понимании принципа работы лазерного принтера.

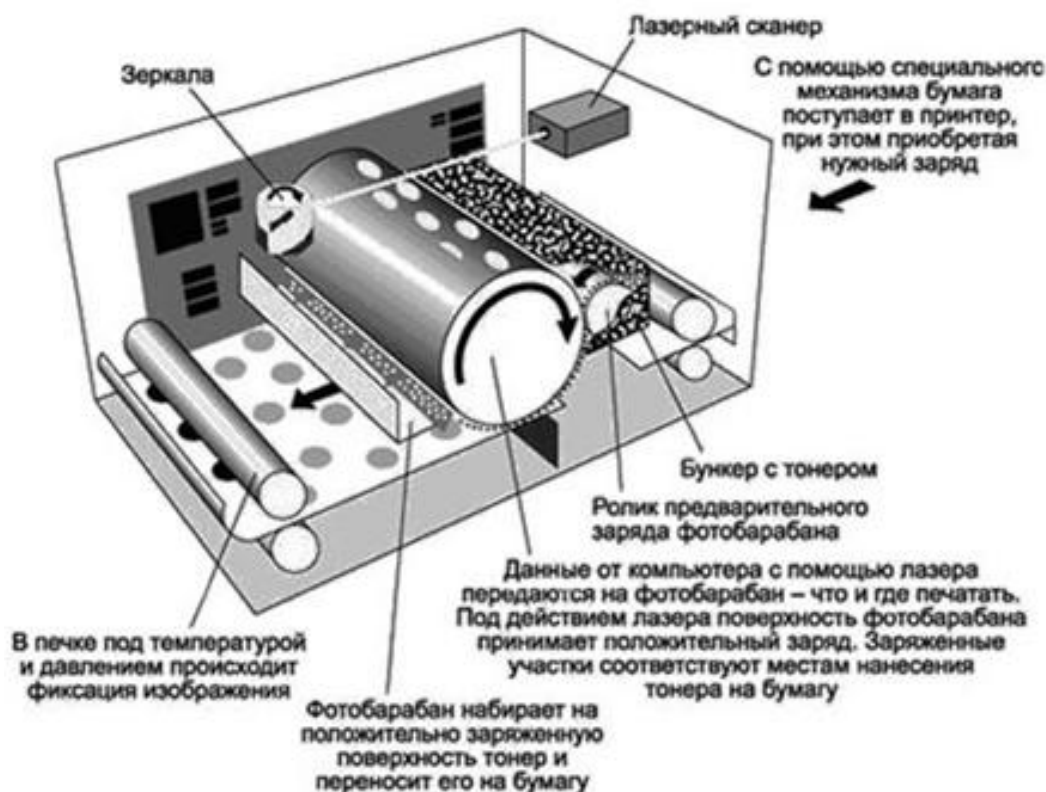


Рисунок 4.37 – Устройство лазерного принтера

Другой важнейшей частью принтера является *лазер* и оптико-механическая система зеркал и линз, перемещающая луч лазера по поверхности барабана. Лазер генерирует очень тонкий световой луч. Отражаясь от вращающихся зеркал, этот луч засвечивает поверхность фото-барабана, снимая ее заряд. Тем самым на поверхность барабана помещается скрытое изображение.

Также обладающий электростатическим зарядом *тонер* (красящий порошок) притягивается к поверхности барабана, сохранившей скрытое изображение. После этого барабан прокатывается по бумаге, и тонер переносится на бумагу. Потом бумага проходит через блок термозакрепления (печку) для фиксации тонера, а фото-барабан очищается от остатков тонера и разряжается.

*Плоттер*. Графопостроитель (от греч. *γράφω* – пишу, рисую) – устройство для автоматического вычерчивания с большой точностью рисунков, схем, сложных чертежей, карт и другой графической информации на бумаге размером до А0 или кальке (рис. 4.38).



Рисунок 4.38 – Плоттер

Графопостроители рисуют изображения с помощью пера (пишущего блока).

Распространенное заблуждение: широкоформатные струйные принтеры иногда неверно называют плоттерами.

*Микрофон и наушники*. Типичный мультимедийный набор включает в себя *наушники* с регулировкой звука, *световой индикатор звука*, мигающий в такт музыки, удобный складывающийся *микрофон* (рис. 4.39).

Конструкция наушников с микрофоном имеет три шнура: один для наушников, второй для микрофона, и третий – это USB-интерфейс, с которого наушники получают обычную электроэнергию для питания сабвуфера. Никаких сигналов операционной системе и от нее при этом не посылая и не получая.



*Рисунок 4.39 – Наушники с микрофоном AudioFX (Audio FX) для компьютера*

*Акустика.* Компьютерная акустика предназначена прежде всего для развлечения пользователя. Встроенного в компьютер динамика хватает ненамного, поэтому очевидно, что наряду с хорошим монитором, удобной мышкой и клавиатурой компьютерные колонки играют очень важную роль в общении пользователя со своим ПК (рис. 4.40). От колонок не менее чем от звуковой карты зависит звучание компакт-диска и качество передачи стереоэффектов в играх.



*Рисунок 4.40 – Пример акустической системы*

Современные акустические системы, за исключением самых простейших, являются многополосными, т. е. в их состав входят динамики нескольких типоразмеров, обычно двух или трех. Каждый воспроизводит свой диапазон частот.

*Источник бесперебойного электропитания (ИБП)* – это автоматическое устройство, основная функция которого – питание нагрузки за счет энергии аккумуляторных батарей при пропадании сетевого

напряжения или выхода его параметров (напряжение, частота) за допустимые пределы (рис. 4.41). Кроме этого, в зависимости от схемы построения ИБП корректирует параметры электропитания.



Рисунок 4.41 – Источник бесперебойного электропитания Powercom KIN 425A

Различают три схемы построения ИБП: резервный (*offline*), интерактивный (*line-interactive*) и он-лайн (*online*).

#### *Многофункциональные периферийные внешние устройства*

В последние годы наметилась тенденция по созданию многофункциональных периферийных внешних устройств. Набор входящих в такой блок устройств может быть разным. Для примера рассмотрим Лазерный принтер/копир/сканер/факс Brother MFC-7420R (рис. 4.42)



Рисунок 4.42 – Многофункциональное внешнее устройство Brother MFC-7420R

Этот аппарат является не просто принтером – он обеспечивает также возможность факсимильной связи, копирования и сканирования, удовлетворяя тем самым любым требованиям современного офиса. В его состав входит:

- принтер: 1200×600 dpi, 20 стр. мин;
- копир: 20 копий/мин., масштабирование 25–400 % с шагом 1 %;
- цветной сканер: 9600 dpi;
- факс лазерный 14400 бит/с, память на 400 стр.

Такое многофункциональное устройство позволяет автоматизировать многие функции небольшого офиса.

### 4.3. Конфигурация компьютера

*Конфигурацией* (или спецификацией) компьютера называют *характеристики устройств*, которые в этот компьютер включены.

Например, в прайс-листе компьютерной фирмы указана такая конфигурация:

Intel Pentium 4 – 3,0 GHz / 512Mb / 120Gb / 128Mb GeForce PCX 6600 / CD-R/RW 52x32x52x / LAN / kbd / M&P / 17» Samsung 710V (LCD, 1280x1024)

Это следует читать так:

- процессор Intel Pentium 4 с тактовой частотой 3,0 гигагерца;
- емкость оперативной памяти – 512 мегабайт;
- жесткий диск (винчестер) емкостью 120 гигабайт;
- графическая плата GeForce PCX 6600 со 128 мегабайтами видеопамяти;
- привод дисков CD, который записывает/перезаписывает/читает диски со скоростью до 52x/32x/52x.
- сетевая плата (LAN);
- клавиатура (*kbd – keyboard*);
- манипулятор мышь и коврик для мыши (*M&P – mouse and pad*);
- жидкокристаллический 17-дюймовый монитор Samsung 710V с «родным» разрешением 1280×1024.

### Вопросы для повторения и самоконтроля

1. На какие пять групп устройств разделены все устройства персонального компьютера?
2. Назначение материнской платы.
3. Назначение центрального процессора.
4. Какие устройства включает центральный процессор?
5. На каком устройстве размещается центральный процессор?

6. С помощью каких устройств осуществляется связь между центральным процессором и остальными устройствами персонального компьютера?

7. Назначение адресной шины и ее разрядность.

8. Назначение шины данных и ее разрядность.

9. Назначение шины команд (управления) и ее разрядность.

10. Какой максимальный объем адресуемой памяти соответствует каждой шине?

11. Что больше: величина адресуемой памяти или размер фактически устанавливаемой оперативной памяти?

12. Как осуществляется подключение отдельных модулей персонального компьютера на физическом и программном уровне?

13. Какие стандартные контроллеры размещаются на системном блоке персонального компьютера?

14. Какой объем данных может одновременно передаваться через последовательный порт?

15. Какой объем данных может одновременно передаваться через параллельный порт?

16. В каком направлении идет изменение рабочего напряжения процессора: увеличения или уменьшения?

17. Какова разрядность процессора современных персональных компьютеров?

18. В чем заключается тактовый принцип исполнения операций персонального компьютера?

19. Чья тактовая частота выше: процессора или материнской платы?

20. На выпуск сколько ядерных процессоров переходят создатели современных компьютеров?

21. Чем известен Гордон Мур?

22. Какие типы памяти входят в состав внутренней памяти?

23. Какое принципиальное отличие между памятью с произвольным доступом и доступом только для чтения?

24. Что такое энергонезависимая и энергозависимая память?

25. Степень какого числа служит основой задания величины емкости памяти?

26. В чем различие между полупроводниковой статической и динамической памятью?

27. Назначение и область применения кэш-памяти.

28. Назначение и область применения постоянной памяти.

29. Назначение и область применения перепрограммируемой постоянной памяти.

30. Что такое BIOS?
31. Что такое CMOS RAM?
32. Что позволяет настраивать программа SETAP?
33. Назначение и принцип работы накопителей на жестких магнитных дисках.
34. Как называется способ записи двоичной информации на жесткие магнитные диски?
35. Какое количество информации записывается в сектор накопителя на жестких магнитных дисках
36. Почему накопители на жестких магнитных дисках носят название «винчестеры»?
37. Действительная емкость накопителя на жестких магнитных дисках маркированного как «100 Мбайт» больше или меньше этой величины?
38. Назначение графической платы (видеокарты).
39. Назначение звуковой платы (звуковой карты).
40. Назначение сетевой платы (сетевой карты).
41. В каких единицах измеряется скорость передачи данных устройствами коммуникации?
42. Что такое НГМД? Имеет ли данный вид носителя перспективы для дальнейшего развития?
43. Назначение и принцип работы накопителей на магнитных дисках.
44. Принцип записи-считывания данных на лазерных дисках.
45. Почему на DVD диск записывается больше информации, чем на CD диск?
46. Флэш-память относится к энергозависимой или энергонезависимой памяти?
47. Назначение и принцип работы клавиатуры. На какие области она обычно делится?
48. Какие типы манипуляторов (координационных устройств) вы знаете?
49. Назначение и принцип работы компьютерной мыши. Способы ее связи с компьютером.
50. Отличие трекбола от мыши.
51. Назначение графического планшета (диджитайзера).
52. Назначение и принцип работы сенсорной панели (тачпада).
53. Назначение и принцип работы джойстика.
54. Назначение и принцип работы сканера. Какие бывают типы сканеров?
55. Назначение и принцип работы цифрового фотоаппарата.



56. Назначение и принцип работы цифровой видеокамеры.
57. Каким образом получается изображение на экране монитора с электронно-лучевой трубкой (CRT)?
58. Что такое «пиксел», «маска» и «частота регенерации» в мониторах типа CRT?
59. В каких единицах измеряется размер монитора? Что такое разрешающая способность монитора?
60. Каким образом получается изображение на экране жидкокристаллического (LSD) монитора?
61. Сделайте сравнительный анализ мониторов CRT и LSD мониторов. Какие из них более удобны?
62. Назначение и принцип работы плазменных панелей (PDP).
63. Какой тип монитора предпочтительней применять в персональных компьютерах: CRT, LSD или PDP и почему?
64. Назначение веб-камер.
65. На какие три основных типа подразделяют принтеры?
66. Принципы работы, преимущества и недостатки матричных принтеров.
67. Принципы работы, преимущества и недостатки струйных принтеров.
68. Принципы работы, преимущества и недостатки лазерных принтеров.
69. Для чего используют плоттер?
70. Назначение модема. Какие типы модемов вам известны?
71. Назначение и принцип работы микрофона.
72. Назначение и принцип работы акустической системы
73. Назначение и принцип работы источников бесперебойного питания.
74. Что такое многофункциональные внешние устройства?

### **Задания для самостоятельной работы**

1. Любая, когда-либо существовавшая вычислительная система обязательно имеет в своем составе (укажите 3 верных ответа):
  - а) центральный процессор;
  - б) звуковую плату;
  - в) оперативную память;
  - г) устройство ввода-вывода;
  - д) винчестер (жесткий диск).

2. Укажите 3 характеристики, относящиеся к процессору:
- а) тактовая частота;
  - б) объем оперативной памяти;
  - в) разрядность;
  - г) объем кэш-памяти.
3. Нагрузка на глаза минимальная при работе с типом монитора:
- а) с CRT-монитором;
  - б) LCD-монитором.
4. Видеопиксель цветного монитора состоит из цветных точек:
- а) белой и черной;
  - б) красной, зеленой, синей;
  - в) голубой, пурпурной, желтой, черной.
5. Из перечисленных устройств для выхода в Интернет применяют:
- а) джойстик;
  - б) модем;
  - в) TV-тюнер.
6. Для нанесения изображения лазерные принтеры используют:
- а) выжигание по бумаге лучом лазера;
  - б) специальный термочувствительный порошок;
  - в) ленту, как у пишущей машинки;
  - г) мелкие капли чернил.
7. При выключении компьютера вся информация стирается:
- а) на гибком диске;
  - б) на CD-диске;
  - в) на жестком диске;
  - г) в оперативной памяти.
8. Выберите неверное утверждение:
- а) в мониторах на жидких кристаллах отсутствует вредное для здоровья электромагнитное излучение;
  - б) процессор относится к внешним (периферийным) устройствам компьютера;
  - в) быстродействие процессора измеряется количеством операций, выполняемых в секунду.
9. Выберите неверное утверждение:
- а) сканер – это устройство, которое чертит графики, рисунки или диаграммы под управлением компьютера;
  - б) накопители на компакт-дисках входят в состав внешней памяти компьютера;
  - в) модем является устройством приема-передачи данных.

*Ответы на задания для самостоятельной работы*

1. Верные ответы: а, в, г
2. Верные ответы: а, в, г
3. Верные ответы: б
4. Верные ответы: б
5. Верные ответы: б
6. Верные ответы: б
7. Верные ответы: г
8. Верные ответы: б
9. Верные ответы: а

## Глава 5. ОРГАНИЗАЦИЯ МЕЖКОМПЬЮТЕРНОЙ СВЯЗИ

Назовем задачи, которые трудно или невозможно решить без организации информационной связи между различными компьютерами:

- перенос информации на большие расстояния (сотни, тысячи километров);
- совместное использование несколькими компьютерами дорогостоящих аппаратных, программных или информационных ресурсов: мощного процессора, емкого накопителя, высокопроизводительного лазерного принтера, баз данных, программного обеспечения и т. д.;
- быстрый перенос информации с одного компьютера на другой;
- совместная работа над большим проектом, когда исполнители должны всегда иметь последние (актуальные) копии общих данных во избежание путаницы и т. д.

### 5.1. Способы организации межкомпьютерной связи

Для того чтобы организовать связь между компьютерами, можно использовать один из следующих способов:

- объединение двух рядом расположенных компьютеров через их коммуникационные порты посредством специального кабеля;
- передача данных от одного компьютера к другому посредством модема с помощью проводных или спутниковых линий связи;
- объединение компьютеров в компьютерную сеть.

Обычно при организации связи между двумя компьютерами за *одним компьютером закрепляется роль поставщика ресурсов* (программ, данных и т. д.), а за *другим – роль пользователя этих ресурсов*. В этом случае первый компьютер называется сервером, а второй – клиентом или рабочей станцией (рис. 5.1).

В некоторых случаях компьютер может быть одновременно и клиентом, и сервером. Это значит, что он может предоставлять свои ресурсы и хранимые данные другим компьютерам и одновременно использовать их ресурсы и данные.

Клиентом также называют прикладную программу, которая от имени пользователя получает услуги сервера. Соответственно, программное обеспечение, которое позволяет компьютеру предоставлять услуги другому компьютеру, называют сервером – так же, как и сам компьютер. Для преодоления несовместимости интерфейсов отдель-

ных компьютеров вырабатывают специальные стандарты, называемые протоколами коммуникации.

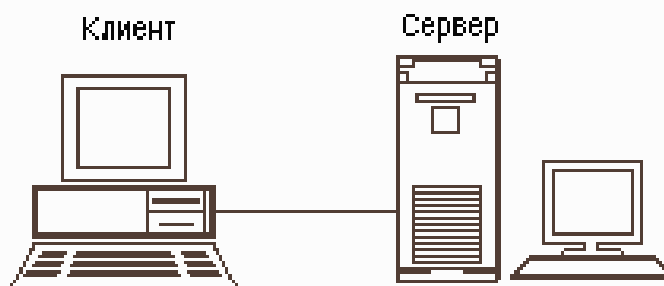


Рисунок 5.1 – Схема соединения компьютеров в сеть по способу «клиент – сервер»

Протокол коммуникации – это согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных. Имеются протоколы для скорости передачи, форматов данных, контроля ошибок и др.

## 5.2. Понятие о компьютерной сети

Компьютерная сеть (англ. *Computer NetWork*, от *net* – сеть и *work* – работа) – совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети.

Ряд основных параметров компьютерных сетей приведен в таблице 5.1.

Таблица 5.1 – Параметры локальной вычислительной сети

Параметр	Локальный (ЛВС / LAN – Local Area Network)	Глобальный (Internet / WAN – Wide Area Network)
Функция	Связывает абонентов одного или нескольких близлежащих зданий одного предприятия	Объединяет абонентов, расположенных по всему миру
Канал передачи данных	Витая пара Оптоволоконный кабель Радиоканал Инфракрасный канал	Оптический кабель Спутниковые каналы
Расстояния между ЭВМ	До 20 км	До 15000 км

Компьютерная сеть представляет совокупность узлов (компьютеров и сетевого оборудования) и соединяющих их ветвей (каналов связи). Ветвь сети – это путь, соединяющий два смежных узла. Различают узлы оконечные, расположенные в конце только одной ветви, промежуточные, расположенные на концах более чем одной ветви, и смежные – такие узлы соединены, по крайней мере, одним путем, не содержащим никаких других узлов. Компьютеры могут объединяться в сеть разными способами.

Логический и физический способы соединения компьютеров, кабелей и других компонентов, в целом составляющих сеть, называется ее топологией. Топология характеризует свойства сетей, не зависящие от их размеров. При этом не учитывается производительность и принцип работы этих объектов, их типы, длины каналов, хотя при проектировании эти факторы очень важны.

Наиболее распространенные три основные топологии сетей:

*Линейная (шинная) сеть.* Содержит только два оконечных узла, любое число промежуточных узлов и имеет только один путь между любыми двумя узлами (рис. 5.2).



Рисунок 5.2 – Топология линейной сети

*Кольцевая сеть.* Сеть, в которой к каждому узлу присоединены две и только две ветви (рис. 5.3). Информация передается последовательно между рабочими станциями до тех пор, пока не будет принята получателем и затем удалена из сети. Недостатком подобной топологии является ее чувствительность к повреждению канала.

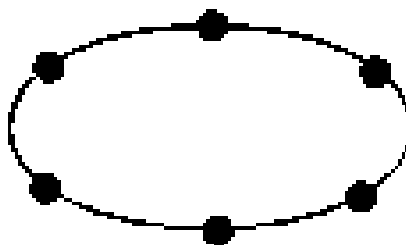


Рисунок 5.3 – Топология кольцевой сети

*Звездообразная сеть.* Сеть, в которой имеется только один промежуточный узел, которому (или через который) посылаются все сообщения (рис. 5.4).

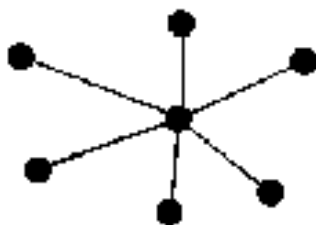


Рисунок 5.4 – Топология звездообразной сети

Важнейшая характеристика компьютерной сети – ее архитектура.

Архитектура сети – это реализованная структура сети передачи данных, определяющая ее топологию, состав устройств и правила их взаимодействия в сети. В рамках архитектуры сети рассматриваются вопросы кодирования информации, ее адресации и передачи, управления потока сообщений, контроля ошибок и анализа работы сети в аварийных ситуациях и при ухудшении характеристик.

Наиболее распространенные архитектуры:

*Ethernet* – широковещательная сеть (скорость передачи данных 10 или 100 Мбит/сек.);

*Arcnet* – широковещательная сеть (скорость передачи данных 2,5 Мбит/сек.);

*Token Ring* – эстафетная кольцевая сеть, сеть с передачей маркера (скорость передачи данных 4 или 16 Мбит/сек.);

*FDDI* – сетевая архитектура высокоскоростной передачи данных по оптоволоконным линиям (скорость передачи – 100 Мбит/сек.);

*ATM* – обеспечивает передачу цифровых данных, видеoinформации и голоса по одним и тем же линиям (скорость передачи до 2,5 Гбит/сек.).

### 5.3. Соединение устройств сети

Для этого используется специальное оборудование:

– сетевые кабели (оптоволоконные; кабели на витых парах, образованные двумя переплетенными друг с другом проводами, и др.);

– коннекторы (соединители) для подключения кабелей к компьютеру; разъемы для соединения отрезков кабеля;

– сетевые интерфейсные адаптеры для приема и передачи данных. В соответствии с определенным протоколом управляют доступом к среде передачи данных. Размещаются в системных блоках компьютеров, подключенных к сети. К разъемам адаптеров подключается сетевая кабель;

- трансиверы повышают уровень качества передачи данных по кабелю, отвечают за прием сигналов из сети и обнаружение конфликтов;
- хабы (концентраторы) и коммутирующие хабы (коммутаторы) расширяют топологические, функциональные и скоростные возможности компьютерных сетей. Хаб с набором разнотипных портов позволяет объединять сегменты сетей с различными кабельными системами. К порту хаба можно подключать как отдельный узел сети, так и другой хаб или сегмент кабеля;
- повторители (репитеры) усиливают сигналы, передаваемые по кабелю при его большой длине.

Локальная (вычислительная) сеть (*ЛВС или LAN – Local Area NetWork*) – сеть, связывающая ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия. Для соединения локальных сетей используют следующие устройства, которые различаются между собой по назначению и возможностям:

- мост (англ. *bridge*) связывает две локальные сети. Передает данные между сетями в пакетном виде, не производя в них никаких изменений. На рис. 5.5 показаны три локальные сети, соединенные двумя мостами. Кроме этого, мосты могут *фильтровать пакеты*, охраняя всю сеть от локальных потоков данных и пропуская наружу только те данные, которые предназначены для других сегментов сети;

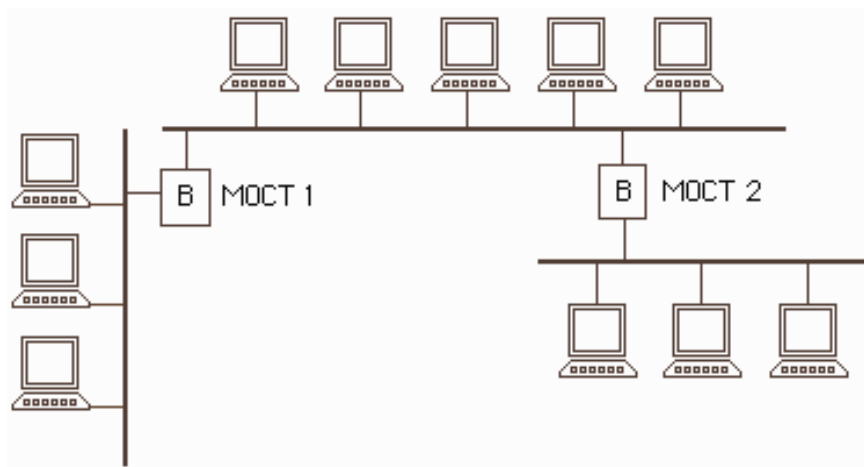


Рисунок 5.5 – Соединение локальных сетей посредством мостов

- маршрутизатор (англ. *router*) объединяет сети с общим протоколом более эффективно, чем мост. Он позволяет, например, расщеплять большие сообщения на более мелкие куски, обеспечивая тем самым взаимодействие локальных сетей с разным размером пакета. Маршрутизатор может пересылать пакеты на конкретный адрес (мос-



ты только отфильтровывают ненужные пакеты), выбирать лучший путь для прохождения пакета и многое другое. Чем сложнее и больше сеть, тем больше выгода от использования маршрутизаторов;

– мостовой маршрутизатор (англ. *brouter*) – это гибрид моста и маршрутизатора, который сначала пытается выполнить маршрутизацию, где это только возможно, а затем, в случае неудачи, переходит в режим моста;

– шлюз (англ. *GateWay*) в отличие от моста, применяется в случаях, когда соединяемые сети имеют различные сетевые протоколы. Поступившее в шлюз сообщение от одной сети преобразуется в другое сообщение, соответствующее требованиям следующей сети. Таким образом, шлюзы не просто соединяют сети, а позволяют им работать как единая сеть. С помощью шлюзов также локальные сети подсоединяются к мэйнфреймам – универсальным мощным компьютерам.

## 5.4. Сеть Интернет

Слово Интернет (*Internet*) происходит от словосочетания *Interconnected networks* (связанные сети) – это глобальное сообщество малых и больших сетей.

Интернет – это информационное пространство, распределенное среди миллионов компьютеров во всем мире, которые постоянно обмениваются данными. Основная задача Интернета – это связь.

Интернет финансируется правительствами, научными и образовательными учреждениями, коммерческими структурами и миллионами частных лиц во всех частях света, но никто конкретно не является ее владельцем. Подключенные к Интернет сети должны удовлетворять определенным стандартам. Эти стандарты утверждаются несколькими добровольными организациями. Например, Совет по архитектуре Интернет (*Internet Architecture Board – IAB*) рассматривает и утверждает протоколы передачи и стандарты нумерации. Комитет по технологическим нормам Интернет устанавливает стандарты повседневной работы сети. Союз Интернет публикует различные стандарты и осуществляет координацию между различными контролирующими органами Интернет, провайдерами услуг и пользователями.

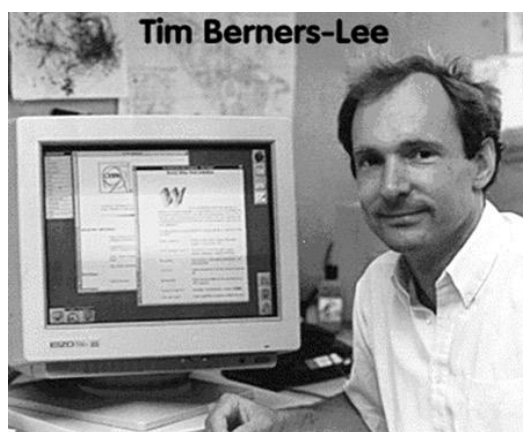
### *История Интернет*

Изобретателями всемирной паутины являются Сэр Тимоти Джон Бёрнерс-Ли (род. 8 июня 1955) и, в меньшей степени, Роберт

Кайо. Тим Бернерс-Ли (рис. 5.6) является автором технологий HTTP, URI/URL и HTML. В 1980 году он работал в Европейском совете по ядерным исследованиям (фр. *Conseil Européen pour la Recherche Nucléaire, CERN*) консультантом по программному обеспечению. Именно там, в Женеве (Швейцария), он для собственных нужд написал программу «Энквайр», которая и заложила концептуальную основу для Всемирной паутины.

В 1989 году, работая в CERN над внутренней сетью организации, Тим Бернерс-Ли предложил глобальный гипертекстовый проект, теперь известный как Всемирная паутина.

В рамках проекта Бернерс-Ли написал первый в мире веб-сервер и первый в мире гипертекстовый веб-браузер, называвшийся *WorldWideWeb*.



*Рисунок 5.6 – Тим Бернерс-Ли*

Первый в мире веб-сайт Бернерс-Ли создал по адресу <http://info.cern.ch>, теперь сайт хранится в архиве. Этот сайт появился в Интернете 6 августа 1991 года. На этом сайте описывалось, что такое Всемирная паутина, как установить веб-сервер, как использовать браузер. Этот сайт также являлся первым в мире Интернет-каталогом, потому что позже Тим Бернерс-Ли разместил и поддерживал там список ссылок на другие сайты.

И все же теоретические основы веба были заложены гораздо раньше. Еще в 1945 году Ваннивер Буш разработал концепцию «Метех» – вспомогательных средств «расширения человеческой памяти». Метех – это устройство, в котором человек хранит все свои книги и записи (а в идеале – и все свои знания, поддающиеся формальному описанию) и которое выдает нужную информацию с достаточной скоростью и гибкостью. Бушем было также предсказано все-

объемлющее индексирование текстов и мультимедийных ресурсов с возможностью быстрого поиска необходимой информации. Следующим значительным шагом на пути к Всемирной паутине было создание гипертекста (термин введен Тедом Нельсоном в 1965 году).

С 1994 года основную работу по развитию Всемирной паутины взял на себя Консорциум Всемирной паутины, основанный Тимом Бернерсом-Ли. W3C – организация, разрабатывающая и внедряющая технологические стандарты для Интернета и Всемирной паутины.

Точные размеры Интернета определить невозможно, поскольку сеть децентрализована и очень подвижна по составу конечных пользователей. Кроме того, принципы и критерии определения числа пользователей Интернета разными экспертами различны, а их результаты часто противоречивы. Одновременно с развитием Интернета увеличивается активность пользователей и время, проведенное в сети. Стремительное развитие и мировая популярность Интернета определяется тем, что эта сеть создала реальную возможность получать и передавать любую информацию кому, где и когда угодно.

Проникновение Интернета в жизнь разных стран не одинаково (табл. 5.2). Так, по данным интернет-сайтов на 2021 год, число пользователей интернета в целом на планете Земля<sup>3</sup> составило 59,5 % от всего населения, а в таких странах как Китай – 71,6 % от населения страны, в России – 85 %, в Индии – 42,5 %.

На конец 2018 года пользователей Интернета в Великобритании – 94,9 % населения, в Германии – 89,7 %, в США – 87,3 %, в Индонезии – 41,1 %, Ангола – 14,3 %, Афганистан – 13,5 % и т. д.

Данные о количестве пользователей Интернет приводятся по состоянию на различные даты и не всегда отражают фактические данные.

Таблица 5.2 – Интернет-пользователи по всему миру, %

	2005	2010	2017*	2019*	2021*
Пользователей по всему миру	16	30	48	53,6	59,5
Пользователей в развивающихся странах	8	21	41,3	47	
Пользователей в развитых странах	51	67	81	86,6	

<sup>3</sup> \* По оценкам. Источник: Международный союз электросвязи. (Список стран по числу пользователей Интернета составлен на основании процента людей в стране, у которых есть доступ к Интернету.) [30]

Пользователи Интернета определяются как лица, которые подключались к Интернету в течение последних 12 месяцев с любого устройства, включая мобильные телефоны.

#### ***5.4.1. Подключения к сети Интернет***

Обращаясь к Интернету, мы пользуемся услугами Интернет-провайдера или *ISP (Internet Service Provider* – поставщик услуг Интернета).

ISP – это организация, которая имеет собственную высокоскоростную сеть, объединенную с другими сетями по всему земному шару. Провайдер подключает к своей сети клиентов, которые становятся частью сети данного провайдера и одновременно частью всех объединенных сетей, которые и составляют Интернет.

Провайдер выступает в качестве посредника (проводника) Интернет, обеспечивая подключение пользователей к Интернет через маршрутизатор Интернет. Пользователь подключается к маршрутизатору провайдера с помощью телефона или выделенной линии.

В зависимости от способа применения существует несколько способов подключения к сети Интернет. Ниже перечислены стандартные способы подключения к Интернет:

- доменный доступ подразумевает, что с провайдером будет заключен договор о возможности непосредственного доступа к Интернет, за которую вы будете вносить месячную или годовую плату. Подобный вид доступа все чаще стал использоваться не только организациями, но и частными лицами;

- клиентский доступ используется для запуска Интернет приложений на рабочих станциях (например, программное обеспечение для торговли акциями, которое связывается с брокерской конторой или коммуникационной программой, проводящей конференцию в режиме реального времени). Подобные приложения самостоятельно устанавливают подключение к Интернет во время запуска и отключаются после завершения работы;

- прямой постоянный доступ используется компаниями, интенсивно предлагающими товары и услуги через Интернет; в качестве примера можно привести авиакомпанию с возможностью бронирования билета через Интернет. Подобный вид доступа является самым дорогим, кроме того, его установка и сопровождение требует дополнительных услуг со стороны провайдера.

Каждый из этих способов подключения предоставляет различный уровень услуг, стоимость подключения при этом различна.

#### **5.4.2. Пересылка данных в Интернет. Протоколы связи TCP/IP**

Отдельные участки Интернет представляют собой сети различной архитектуры, которые связываются между собой с помощью маршрутизаторов. Передаваемые данные разбиваются на небольшие порции, называемые пакетами. Каждый пакет перемещается по сети независимо от других пакетов. Сети в Интернет неограниченно коммутируются (т. е. связываются) друг с другом, потому что все компьютеры, участвующие в передаче данных, используют единый протокол коммуникации TCP/IP. На самом деле протокол TCP/IP – это два разных протокола, определяющих различные аспекты передачи данных в сети:

– протокол TCP (*Transmission Control Protocol*) – протокол управления передачей данных, использующий автоматическую повторную передачу пакетов, содержащих ошибки; этот протокол отвечает за разбиение передаваемой информации на пакеты и правильное восстановление информации из пакетов получателя;

– протокол IP (*Internet Protocol*) – протокол межсетевого взаимодействия, отвечающий за адресацию и позволяющий пакету на пути к конечному пункту назначения проходить по нескольким сетям.

Схема передачи информации по протоколу TCP/IP такова: протокол TCP разбивает информацию на пакеты – отдельные блоки фиксированного размера, и нумерует все пакеты, чтобы их затем можно было собрать в правильной последовательности. К данным, содержащимся в пакете, добавляется дополнительная информация примерно такого формата.

Адрес получателя	Адрес отправителя	Длина	Данные	Поле контрольной суммы
------------------	-------------------	-------	--------	------------------------

Контрольная сумма данных пакета содержит информацию, необходимую для контроля ошибок. Первый раз она вычисляется передающим компьютером. После того, как пакет будет передан, контрольная сумма повторно вычисляется принимающим компьютером. Если значения не совпадают, это означает, что данные пакета были повреждены при передаче. Такой пакет отбрасывается, и автоматически направляется запрос.

Далее с помощью протокола IP все пакеты передаются получателю, где с помощью протокола TCP проверяется, все ли пакеты получены; после получения всех пакетов протокол TCP располагает их в нужном порядке и собирает в единое целое.

### 5.4.3. Адресация в Интернете

**IP-адресация.** Чтобы компьютеры, объединенные в сеть, могли обмениваться сообщениями, каждый из них должен иметь *уникальный* адрес. В сети Интернет это *32-разрядный* (т. е. 32-битный = 4-байтный) *адрес*, называемый IP-адрес (табл. 5.3).

Таблица 5.3 – Пример написания IP-адреса

IP-адрес двоичный	11011100	11010111	00001110	00010110
IP-адрес десятичный	220	215	14	22

В точечно-десятичной нотации IP-адрес может выглядеть, например, так: 220.215.14.22. Каждая часть, разделенная точкой, представляет собой один байт, и следовательно, максимальное десятичное число, которое может быть представлено одним байтом 255 ( $2^8 = 256$ , от 0 до 255).

Для человека такая система адресации сложна, так же, как нам сложно помнить, набирать и диктовать одиннадцатизначные телефонные номера, поэтому в 1984 году Полом Мокапетрисом была разработана *надстройка над IP-адресацией*, называемая системой *DNS* (*domain name system, система доменных имен*).

#### **DNS – система доменных имен**

Доменные имена системы DNS – синонимы IP-адреса. Они символьные, а не числовые; они удобнее для запоминания и ориентации; они несут смысловую нагрузку.

www.irnet.ru -> таблицы DNS -> 193.232.70.36

Доменные имена также уникальны, т. е. нет в мире двух одинаковых доменных имен. Доменные имена, в отличие от IP-адресов не обязательны, они приобретаются дополнительно.

Также уникальны адреса, которые указывают на конвертах при доставке писем обычной почтой. В мире нет стран с одинаковыми на-

званиями. И если названия городов иногда и повторяются, то в сочетании с делением на более крупные административные единицы типа районов и областей они становятся уникальными. А названия улиц не должны повторяться в пределах одного города. Таким образом, адрес на основе географических и административных названий однозначно определяет точку назначения.

Домены имеют аналогичную иерархию. Имена доменов отделяются друг от друга точками: `lingvo.yandex.ru`, `krkime.com` (рис. 5.7).

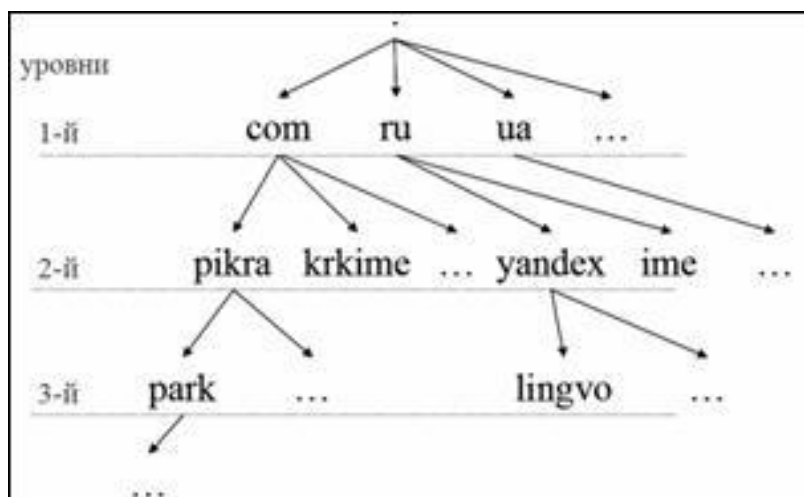


Рисунок 5.7 – Иерархия в системе DNS

Домены первого уровня разделяются на *тематические* (табл. 5.4) и *географические*.

Таблица 5.4 – Имена доменов первого уровня

Административный домен	Тип организации	Географический домен	Страна
com	Коммерческая	ca	Канада
edu	Образовательная	de	Германия
gov	Правительственная США	jp	Япония
int	Международная	.ru и .рф	Россия
mil	Военная США	.by и .бел	Белоруссия
net	Компьютерная сеть	uk	Англия
org	Некоммерческая	us	США

Регистрация доменного имени второго уровня в тематических доменных зонах доступна для организации или частного лица независимо от географического положения. Стоимость владения доменным

именем в такой зоне не превышает \$35 в год. Юридического оформления владения не требуется, требуется только выбрать подходящее доменное имя из числа не занятых и перечислить деньги. Одна из организаций, осуществляющих регистрацию имен в этих доменных зонах – Network Solution (<http://www.netsol.com>).

Национальные (географические) домены верхнего уровня всегда двухбуквенные.

Регистрация доменных имен второго уровня в зонах .RU и .РФ<sup>4</sup> производится организацией RU-CENTER (<http://www.nic.ru>). При регистрации требуется юридическое оформление договорных отношений, которое занимает около месяца, с учетом пересылки документов в Москву и из Москвы.

Самые популярные геодомены: .msk.ru – Москва, .spb.ru – Санкт-Петербург, .nov.ru – Великий Новгород, всего же их более 60.

В геодоменах большое количество свободных «красивых» имен – легко подобрать короткое и запоминающееся имя для сайта. Кроме того, стоимость регистрации и продления геодоменов ниже по сравнению с другими зонами.

Существует ряд доменов третьего уровня, которые не имеют географической привязки, они указывают на отношение сайта к российскому сегменту интернета:

.com.ru – домен для коммерческих организаций, ведущих деятельность на территории РФ.

.exnet.ru – домен для размещения сетевых приложений (excellence network).

.ru.net – домен для сайтов, связанных с сетевыми технологиями и развитием Рунета.

.pp.ru – домен для персональных страниц и проектов.

.net.ru – домен для проектов, связанных с веб-технологиями.

.org.ru – домен для российских организаций, в том числе некоммерческих.

### ***Система адресации URL***

Чтобы найти в Интернете какой-либо документ, достаточно знать *ссылку* на него – так называемый универсальный указатель ресурса (*URL – Uniform Resource Locator*), который определяет место-

---

<sup>4</sup>.рф – национальный домен верхнего уровня для России. Введен 12 мая 2010 года. Первый в Интернете домен на кириллице. Отличием от введенного ранее домена «.ru» является то, что в домене «.рф» все имена второго уровня пишутся исключительно кириллицей.



нахождение каждого файла, хранящегося на компьютере, который подключен к Интернету.

Адрес URL является сетевым расширением понятия полного имени ресурса в операционной системе. В URL, кроме имени файла и директории, где он находится, указывается сетевое имя компьютера, на котором этот ресурс расположен, и протокол доступа к ресурсу, который можно использовать для обращения к нему. Система адресации URL и адресация почтовой службы имеют сходную структуру.

Проведем аналогию с доставкой обычного письма в адрес некоторой организации (например, института) на имя конкретного человека (рис. 5.8).

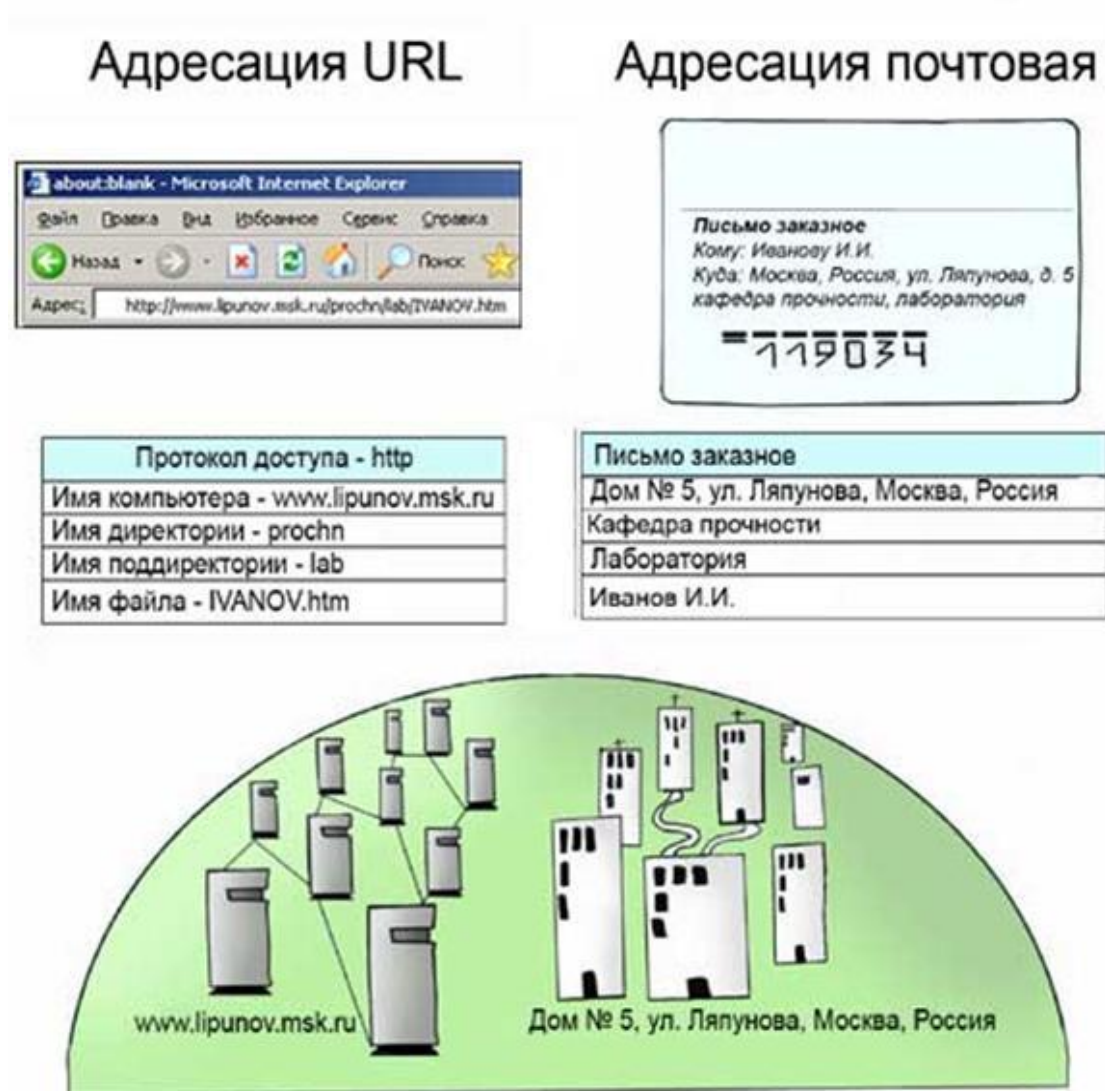


Рисунок 5.8 – Сравнение адресации URL с почтовой адресацией

Рассмотрим структуру следующего URL адреса:  
<http://www.lipunov.msk.ru/prochn/lab/IVANOV.htm> .

Первая часть *http://* (HyperText Transfer Protocol – протокол передачи гипертекста, по которому обеспечивается доставка документа с Web-сервера Web-браузеру) указывает программе просмотра (браузеру), что для доступа к ресурсу применяется данный сетевой протокол.

В URL первым стоит указатель на тип доступа к запрашиваемому файлу, а затем его адрес.

Вторая часть *www.lipunov.msk.ru* указывает на доменное имя и адресует конкретный компьютер.

Третья часть *prochn/lab/IVANOV.htm* показывает программе-клиенту, где на данном компьютере-сервере искать ресурс.

В рассматриваемом случае ресурсом является файл в формате *html*, а именно *IVANOV.htm*, который находится в папке *lab*, которая в свою очередь расположена в папке *prochn*.

Имена каталогов, содержащиеся в URL, обычно являются виртуальными и не имеют ничего общего с реальными именами каталогов компьютера, на котором выполняется Web-сервер, а являются их псевдонимами: ни один владелец компьютера, на котором выполняется Web-сервер, не позволит постороннему пользователю, обращающемуся к Web-серверу через Интернет, получить доступ к реальной файловой системе этого компьютера.

Обратите внимание: *при написании URL важно различать прописные и строчные буквы.*

#### ***5.4.4. Обзор сервисов Интернета***

Часто путают два понятия – Интернет и WWW (или Web).

Следует напомнить, что WWW (Всемирная паутина World Wide Web) – это лишь одна из многочисленных услуг (сервисов), предоставляемых пользователям Интернета.

На базе Интернета может быть реализовано множество информационных служб, с помощью которых пользователь Интернета может получать различные сервисы. В таблице 5.5 приведены некоторые, широко используемые сервисы обмена данными на базе сети Интернет.

Предоставляемые сетью Интернет сервисы обладают различной степенью динамичности и персонализации.

World Wide Web (WWW, Всемирная паутина) – гипертекстовая, а точнее гипермедийная информационная система поиска ресурсов Интернет и доступа к ним.

Таблица 5.5 – Сервисы обмена данными на базе сети Интернет

Сервис	Назначение
E-mail mailto:max@mail.ru	Позволяет обмениваться текстовыми сообщениями, к которым присоединяются файлы любых типов
ICQ UIN: 189764452	Интернет-пейджер. Служит для персонального интерактивного общения в режиме реального времени
FTP ftp://ftp.museum.ru	File Transfer Protocol - протокол передачи файлов. Служит для перемещения файлов между компьютерами сети Интернет
Telnet telnet 10.100.254.254	Позволяет подключаться по Интернет к удаленному компьютеру и работать с ним так, как будто вы находитесь за его терминалом
WWW http://www.krkime.com	Обеспечивает доступ к гигантскому объему информации: текст, графика, аудио, видео, программы. В основе – технология гипертекста

**Гипертекст** – информационная структура, позволяющая устанавливать смысловые связи между элементами текста на экране компьютера таким образом, чтобы можно было легко осуществлять переходы от одного элемента к другому. На практике в гипертексте некоторые слова выделяют путем подчеркивания или окрашивания в другой цвет. Выделение слова говорит о наличии связи этого слова с некоторым документом, в котором тема, связанная с выделенным словом, рассматривается более подробно.

**Гипермедиа** – то, что получится, если в определении гипертекста заменить слово «текст» на «любые виды информации»: звук, графику, видео. Такие гипермедийные ссылки возможны, поскольку наряду с текстовой информацией можно связывать и любую другую двоичную информацию, например, закодированный звук или графику. Так, если программа отображает карту мира и если пользователь выбирает на этой карте с помощью мыши какой-либо континент, программа может тут же дать о нем графическую, звуковую и текстовую информацию.

Система WWW построена на специальном протоколе передачи данных, который называется протоколом передачи гипертекста **НТТР** (*HyperText Transfer Protocol*). Все содержимое системы WWW состоит из WWW-страниц.

**WWW-страницы** – гипермедийные документы системы World Wide Web. Создаются с помощью языка разметки гипертекста HTML (*Hypertext markup language*).

Язык HTML позволяет добавлять к текстовым документам специальные командные фрагменты – тэги (англ. *tag* – этикетка, ярлык) таким образом, что становится возможным связывать с этими документами другие тексты, графику, звук и видео, задавать заголовки различных уровней, разделять текст на абзацы, строить таблицы и т. д.

Одну WWW-страницу на самом деле обычно составляет *набор гипермедийных документов, расположенных на одном сервере, переплетенных взаимными ссылками и связанных по смыслу*. Каждый документ страницы, в свою очередь, может содержать несколько экранных страниц текста и иллюстраций. Каждая WWW-страница имеет свой титульный лист (англ. *homepage*) – гипермедийный документ, содержащий ссылки на главные составные части страницы. Адреса титульных листов распространяются в Интернете в качестве адресов страниц.

Личные страницы – такие WWW-страницы, которые принадлежат не фирмам и не организациям, а отдельным людям. Содержание и оформление такой страницы зависит только от ее автора.

При работе с системой WWW пользователи имеют дело с *программами-клиентами системы, называемыми браузерами*.

**Браузеры** (англ. *browse* – листать, просматривать) – программы, с помощью которых пользователь организует диалог с системой WWW: просматривает WWW страницы, взаимодействует с WWW-серверами и другими ресурсами в Интернет.

Браузеры WWW умеют взаимодействовать с любыми типами серверов, используя при этом их собственные протоколы. Информацию, полученную от любого сервера, браузер WWW выводит на экран в стандартной, удобной для восприятия форме. При этом переключения с одного протокола на другой для пользователя часто остаются незамеченными.

Существуют сотни программ-браузеров. По данным сервиса StatCounter<sup>5</sup> (на октябрь 2021 г.) на международном рынке браузеров как для мобильных устройств, так и для компьютеров, браузер Google Chrome продолжает доминировать в международном масштабе, занимая первое место с долей рынка 64,7 %, второе и третье места за-

---

<sup>5</sup>StatCounter – <https://www.ixbt.com/news/2021/11/01/nazvany-samye-populjarnye-brauzery-v-mire--v-rossii-kartina-silno-otlichaetsja-ot-obsheirovoj.html> (дата обращения 20.05.2022)[34].

нимают Safari от Apple с 19,04 % и Microsoft Edge с 3,99 %. На четвертом месте с небольшим отрывом расположился Firefox с 3,67 %. Пятое место занимает мобильный браузер Samsung Internet с долей 2,8 %, который мы можем увидеть на смартфонах и планшетах Samsung. А браузеру Opera досталась доля в 2,36 %.

Что касается России, здесь картина несколько иная. На первом месте, конечно, находится Chrome, но с заметно меньшей долей, а второе место уверенно занимает Яндекс.Браузер.

Топ-6 браузеров StatCounter для России выглядит так: Chrome – 56,94 %, Яндекс.Браузер – 14,25 %, Safari – 9,67 %, Opera – 8,35 %, Firefox – 4,46 %, Samsung Internet – 2,51 %.

**Электронная почта** (*Electronic mail*, англ. *mail* – почта, сокращенно *e-mail*) служит для передачи текстовых сообщений в пределах Интернет, а также между другими сетями электронной почты. К тексту письма современные почтовые программы позволяют прикреплять звуковые и графические файлы, а также двоичные файлы – программы. При использовании электронной почты каждому абоненту присваивается уникальный почтовый адрес, формат которого имеет вид: <имя пользователя> @ <имя почтового сервера>.

Например: *earth@space.com*, где *earth* – имя пользователя, *space.com* – имя компьютера, @ – разделительный символ «эт коммерческое».

Сообщения, поступающие по e-mail, хранятся в специальном почтовом компьютере в выделенной для получателя области дисковой памяти (его почтовом ящике), откуда он может их выгрузить и прочитать с помощью специальной программы-клиента. Для отсылки сообщения нужно знать электронный адрес абонента. При качественной связи электронное письмо доходит в любую точку мира в течение нескольких минут.

Пользователи электронной почты стремятся придерживаться правил сетевого этикета (нэтикета), а для выражения эмоций используют схематические изображения человеческого лица, так называемые смайлики (англ. *smile*, «улыбка»).

### **Поиск во Всемирной паутине Интернет**

В Web размещены миллионы сайтов, причем с актуальной информацией соседствует много устаревших ресурсов, мусора и недобросовестной рекламы.

Как росло количество веб-сайтов в мире (рис. 5.9)<sup>6</sup>.

## КАК РОСЛО КОЛИЧЕСТВО ВЕБ-САЙТОВ В МИРЕ

ИСТОЧНИК: INTERNET LIVE STATS.

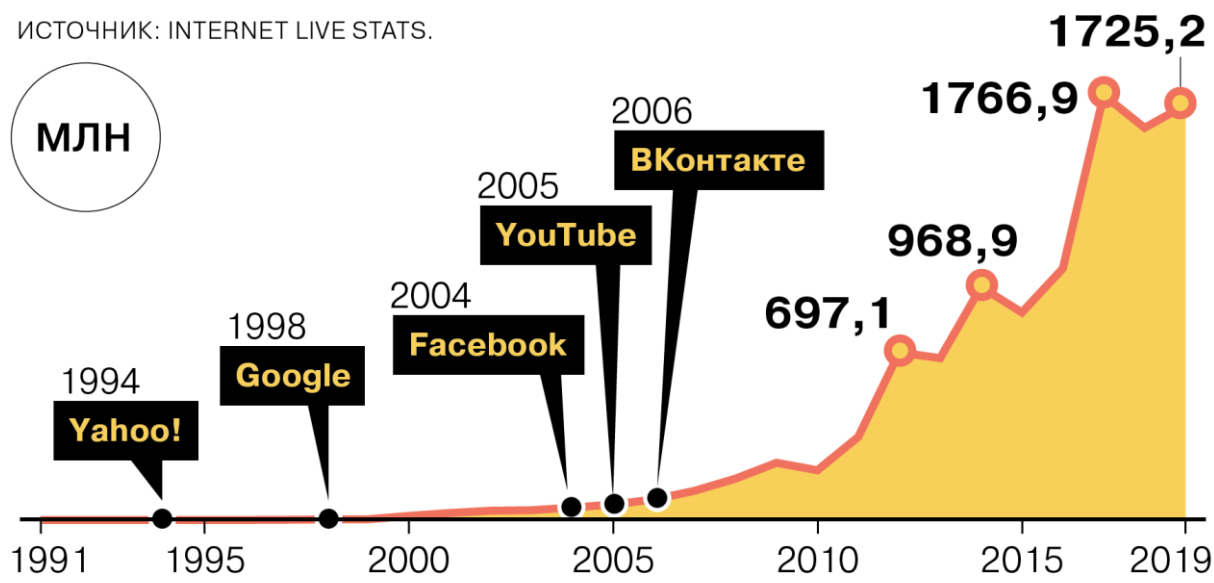


Рисунок 5.9 – Статистика роста количества веб-сайтов

В августе 1991 года исследователь из Европейского центра ядерных исследований (ЦЕРН) Тим-Бернерс Ли запустил первый в истории веб-сайт, который существует и по сей день. К концу 1992 года количество сайтов выросло до 10, а после того как технология *WorldWideWeb* (W3) стала общедоступной в 1993 году, интернет начал стремительно расти, что привело к глобальным переменам по всему миру. К концу 1994 года всемирная паутина насчитывала 2,7 тыс. сайтов. Тогда же на свет появилась одна из первых поисковых систем – «Путеводитель Джерри по Всемирной паутине», которая впоследствии была переименована в «Yahoo!». Уже в 1995 году в интернете работало 23,5 тыс. веб-сайтов, в 1996 году – почти в десять раз больше. На момент регистрации компании Google в 1998 году количество сайтов достигло отметки в 2,4 млн. В 2001 году работало уже 29,2 млн веб-сайтов. В том же году была запущена универсальная интернет-энциклопедия Wikipedia. В 2006 году в Интернете заработала российская социальная сеть «ВКонтакте», тогда всемирная паутина насчитывала уже 85,5 млн сайтов. Рубеж в 1 млрд был пройден в 2016 году, а уже в 2017 году общее число веб-сайтов выросло до 1,7 млрд. С этого момента их количество практически не измени-

<sup>6</sup>По данным сайта <https://www.kommersant.ru/doc/4147760> (дата обращения 20.05.2022) [33].

лось, и к настоящему времени общее число сайтов в мире составляет чуть более 1,7 млрд.

Интернет – это наиболее демократичный источник информации. Каждый может разместить в Сети собственный ресурс и высказать свое мнение. В этом одновременно сила и слабость Всемирной сети.

Находить информацию в Интернете, вероятно, было бы очень трудно, если бы не были созданы мощные поисковые инструменты: поисковые машины (поисковики), каталоги-рейтинги (рубрикаторы), тематические списки ссылок, онлайн-энциклопедии и словари.

Для поиска разного рода информации наиболее эффективными оказываются различные инструменты.

### **Каталоги ресурсов**

Каталог имеет иерархическую структуру. Тематические разделы первого уровня определяют максимально широкие темы: спорт, отдых, наука, магазины и т. д. В каждом таком разделе могут быть подразделы. Пользователь может уточнять интересующую его область, путешествуя по дереву каталога и постепенно сужая зону поиска. Например, при поиске информации о ноутбуках цепочка поиска может выглядеть так: информационные технологии – компьютеры – ноутбуки. Дойдя до нужного подкаталога, пользователь находит в нем набор ссылок.

Обычно в каталоге все ссылки являются профильными, поскольку составлением каталогов занимаются не программы, а люди. Очевидно, что если ведется поиск общей информации по некоторой широкой теме, то целесообразно обратиться к каталогу. Если же необходимо найти конкретный документ, то каталог окажется малоэффективным поисковым средством.

Часто каталоги ресурсов одновременно являются и рейтингами, т. е. каталог предлагает зарегистрированным в нем сайтам установить на своих страницах *счетчик посещений*, и отображает списки ссылок на сайты в соответствии с их популярностью (посещаемостью). Популярность ресурса оценивается по ряду параметров, в том числе по так называемым *хостам* (количество уникальных посетителей в сутки) и *хитам* (количество заходов на сайт в сутки).

Одним из наиболее популярных каталогов-рейтингов является Rambler's Top 100. (<http://top100.rambler.ru/top100/>). Часто бывает интересно оценить состояние не общероссийских, а региональных ресурсов по конкретной тематике. Для обзора web-ресурсов Краснояр-

ска и края можно рекомендовать каталоги-рейтинги ресурсов Krasland (<http://www.krasland.ru/>) и Сталкер (<http://www.stalker.internet.ru/>).

### **Поисковые машины**

*Релевантный документ* – документ, смысловое содержание которого соответствует *информационному запросу*. Современные поисковые машины осуществляют поиск по контексту, т. е. словам, содержащимся в запросе, учитывая вариации словоформ и расширяя запросы синонимами. Но смысла компьютеры не понимают, поэтому в списке ответов на запрос, наряду с релевантными вашему запросу документами, вы можете получить и те, которые вам не подходят.

Очевидно, что *от умения грамотно выдавать запрос зависит процент получаемых релевантных документов*. Доля релевантных документов в списке всех найденных поисковой машиной называется точностью поиска. Нерелевантные документы называют шумовыми. Если все найденные документы релевантные (шумовых нет), то точность поиска составляет 100 %. Если найдены все релевантные документы, то полнота поиска – 100 %.

Таким образом, *качество поиска определяется двумя взаимозависимыми параметрами: точностью и полнотой поиска. Увеличение полноты поиска снижает точность, и наоборот*.

Поисковые системы можно сравнить со справочной службой, агенты которой обходят предприятия, собирая информацию в базу данных. При обращении в службу информация выдается из этой базы. Данные в базе устаревают, поэтому агенты их периодически обновляют. Иными словами, справочная служба имеет две функции: первая – создание и постоянное обновление данных в базе и вторая – поиск информации в базе по запросу клиента.

Аналогично, поисковая машина состоит из двух частей: так называемого *поискового робота* (или паука), который обходит серверы Сети и формирует базу данных, и *механизма поиска* релевантных запросу пользователя ссылок в базе.

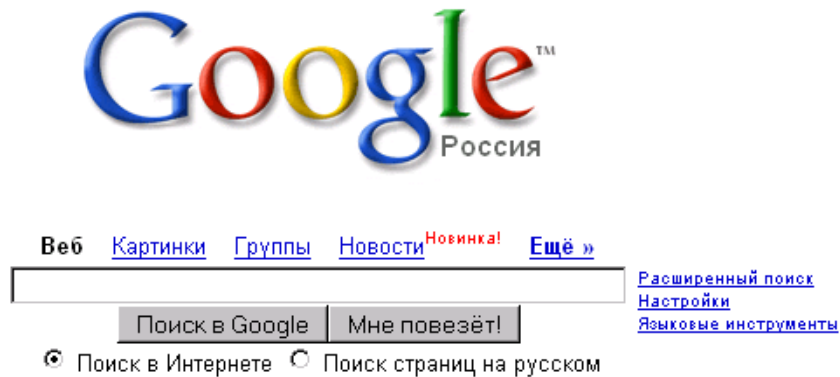
Следует отметить, что, отрабатывая конкретный запрос пользователя, поисковая система оперирует именно внутренней базой данных (а не пускается в путешествие по Сети). Несмотря на то, что база данных поисковой машины постоянно обновляется, поисковая машина не может проиндексировать все Web-документы: их число слишком велико. Проблема недостаточности полноты поиска состоит не только в ограниченности внутренних ресурсов поисковика, но и в



том, что скорость работа ограничена, а количество новых Web-документов постоянно растет.

Наиболее популярными на сегодня поисковыми системами являются Google ([www.google.com](http://www.google.com), [www.google.ru](http://www.google.ru)) – рис. 5.10 и Яндекс ([www.yandex.ru](http://www.yandex.ru)) – рис. 5.11.

[Моя страница iGoogle](#) | [Войти](#)



[Рекламные программы](#) - [Всё о Google](#) - [Работа в Google](#) - [Google.com in English](#)

©2007 Google

Рисунок 5.10 – Главная страница поисковой системы Google

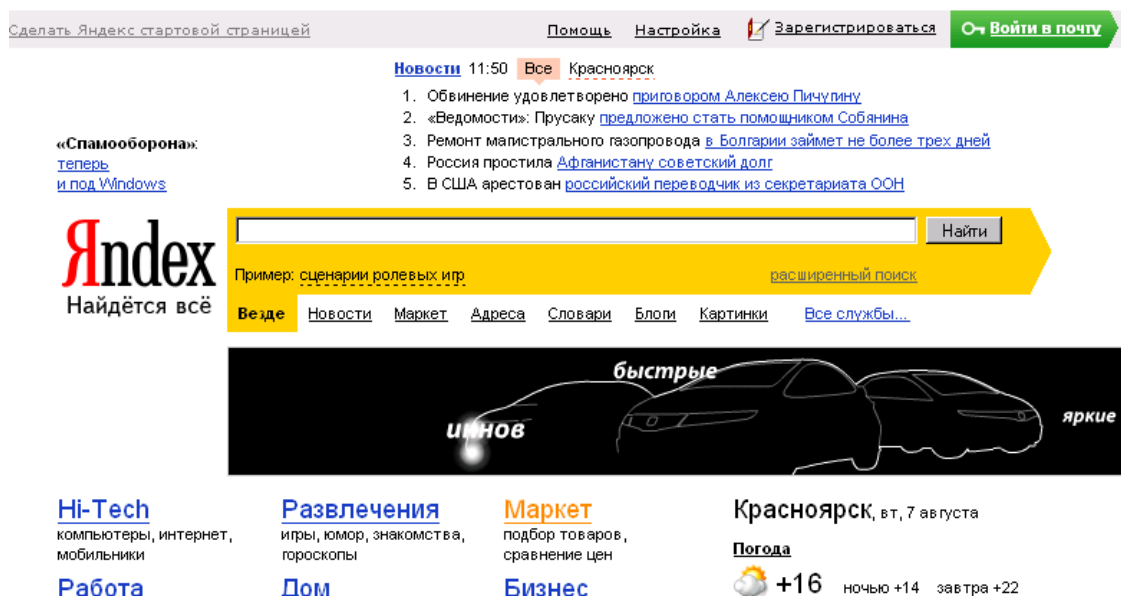


Рисунок 5.11 – Главная страница поисковой системы Яндекс

## Онлайновые энциклопедии и справочники

В ряде случаев бывает нужно найти не просто документ, содержащий ключевое слово, а именно толкование слова. При поиске незнакомого термина с помощью поисковой машины вы рискуете полу-

чить целый ряд статей, в которых этот термин используется, и при этом так и не узнать, что же он все-таки обозначает. Подобный поиск предпочтительнее проводить в онлайн-энциклопедии.

Одной из крупнейших онлайн-энциклопедий является ресурс «Яндекс. Энциклопедии» (<http://encycl.yandex.ru>) – этот проект содержит 14 энциклопедий, в том числе статьи из Большой Советской Энциклопедии и Энциклопедию Брокгауза и Эфрона (рис. 5.12). При написании данного пособия использовался Энциклопедический систематизированный словарь-справочник по информатике (<http://encycl.yandex.ru/dict/informatica>). К крупным относится и «Энциклопедия Кирилла и Мефодия» (<http://www.km.ru>).

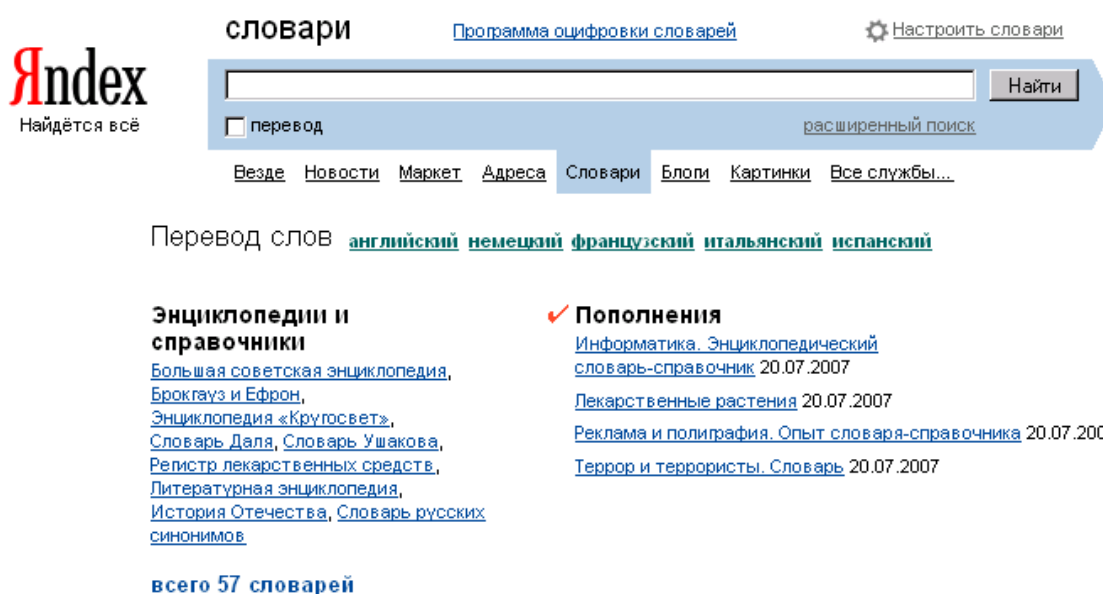


Рисунок 5.12 – Главная страница онлайн-энциклопедий Яндекс

Помимо переноса в гипертекстовую среду традиционных словарей бурно развиваются энциклопедические **wiki-проекты**.

**Вики** – веб-сайт для сбора и структуризации письменных сведений. Характеризуется тем, что наполнять и редактировать размещаемую на нем информацию могут все посетители.

<http://ru.wikipedia.org> – википедия на русском языке – часть многоязычного проекта, целью которого является создание полной энциклопедии на всех языках Земли.

**Программа пересылки файлов FTP.** Перемещает копии файлов с одного узла Интернет на другой в соответствии с протоколом FTP (*File Transfer Protocol* – «протокол передачи файлов»). При этом не имеет значения, где эти узлы расположены и как соединены между

собой. Компьютеры, на которых есть файлы для общего пользования, называются FTP-серверами. В Интернет имеется более 10 Терабайт бесплатных файлов и программ.

### **Web-сайт организации**

Различные бизнесы по-разному используют возможности сети Интернет и web-технологий. В некоторых случаях возможен полный перенос деятельности в Сеть. Например, открытие Интернет-магазинов или публикация рекламным агентством баз данных соискателей и вакансий на своем сайте в Интернет, для того, чтобы работодатели и соискатели самостоятельно выполняли всю работу по подбору вариантов, и обращались в агентство только за недостающей контактной информацией, которая и является в этом случае товаром.

Но любая организация, даже очень далекая по роду деятельности от информационных технологий может использовать собственный web-сайт как виртуальную выставку своих товаров и/или услуг, доступную заинтересованным лицам 24 часа в сутки 7 дней в неделю без территориальных ограничений.

Даже простейший сайт, так называемый «сайт-визитка», решает представительские задачи организации и помогает разгрузить информационные потоки в офисе, за счет того, что на нем размещаются прайс-листы, формы договоров, ответы на типичные вопросы клиентов, адресная и контактная информация, организуется online-прием заказов и т. п.

Современная информационная культура приучила многих потенциальных клиентов сначала выполнять поиск необходимых товаров и услуг в Интернет, анализировать предложения и, обладая необходимой информацией и компетентностью, выступать в качестве заказчика.

Web-сайт организации должен производить приятное впечатление на потенциальных клиентов и партнеров, поэтому важную роль играет его дизайн, основанный на фирменном стиле организации. Разработку такого web-сайта целесообразно проводить силами штатных IT-специалистов, даже если html-разметка документа не представляет для них проблемы. Эффективнее обратиться в специализированную организацию – web-студию.

Создания web-сайта организации включает следующие *этапы*:

1. Определение целей и задач сайта, составление примерной информационной структуры, создание контента (информационного наполнения) сайта.

2. Заказ сайта и документации по его администрированию web-студии.
3. Определение адреса сайта (покупка доменного имени).
4. Размещение сайта на web-сервере (хостинг: на собственном интернет-канале или у провайдера, занимающегося хостингом).
5. Назначение и обучение специалиста, ответственного за сопровождение сайта.
6. Включение адреса сайта во все рекламные материалы организации.
7. Рекламирование сайта в сети Интернет (регистрация в поисковых системах и каталогах, включение в баннерный обмен, рекламирование на досках объявлений.). *Banner* (флаг, транспарант) – статическая или анимированная картинка одного из стандартных форматов (468×60, 120×60, 100×100, 88×31 pix), размещаемая на web-страницах с рекламной целью. Является ссылкой на рекламируемый ресурс. Рекламирование сайта тоже обычно поручается web-студии.

### **Internet-торговля**

Организация интернет-торговли. Как правило, когда говорят про использование технологий Интернета при организации торговли, то выделяют отдельное направление бизнеса, называемое электронной коммерцией. Особенности предоставления услуг и реализации товаров через Интернет таковы, что позволяют нам говорить об этом виде организации бизнеса отдельно.

Суть интернет-коммерции в предоставлении услуг через сеть Интернет. При этом такая форма бизнеса может быть как дополнительной к основному бизнесу, так и самостоятельным бизнесом. Посредством сети может предоставляться и продаваться информация, товары, информация об услугах и т. д. Перспективность электронного бизнеса связана, в первую очередь, с сокращением издержек при переходе на эту форму организации бизнеса, увеличением оборачиваемости и увеличением объемов реализации с ростом аудитории Интернета.

В последние годы наблюдается активное развитие электронной коммерции. Как отмечалось ранее, в настоящее время, свыше 1 млрд во всем мире и около 30 млн человек в России, имеют доступ к Интернету. И поверьте нам, это не самые бедные люди.

### **Эффективность интернет-торговли**

Благодаря низким барьерам проникновения на рынок численность интернет-магазинов растет интенсивно. Однако рентабельность

интернет-торговли достаточно низка, эффективная внутренняя норма доходности бизнеса обеспечивается, прежде всего, интенсивным оборотом и минимизацией издержек. Основным препятствием, на наш взгляд, является не столько задача формирования интернет-витрины, сколько сложность достижения необходимого оборота. Затраты на рекламу интернет-ресурса достаточно велики. Также значительной статьей издержек является сбор и публикация околонижной информации на сайте. Последнее необходимо для привлечения интереса клиентов к сайту не только как к интернет-магазину, но и как к информационно-развлекательному ресурсу. Интенсивный рост числа магазинов сдерживается этими факторами, многие участники выбывают или оказываются поглощенными более эффективными и агрессивными конкурентами.

### Организация интернет-магазина

Для покупателя работа с интернет-магазином схематично выглядит следующим образом (рис. 5.13). Посетитель через компьютер, подключенный к Интернету, выходит на сайт магазина, выбирает необходимый товар, формирует на него заказ, указывает свои персональные данные. После чего в подтверждение обработки заказа он получает письмо на свою электронную почту.

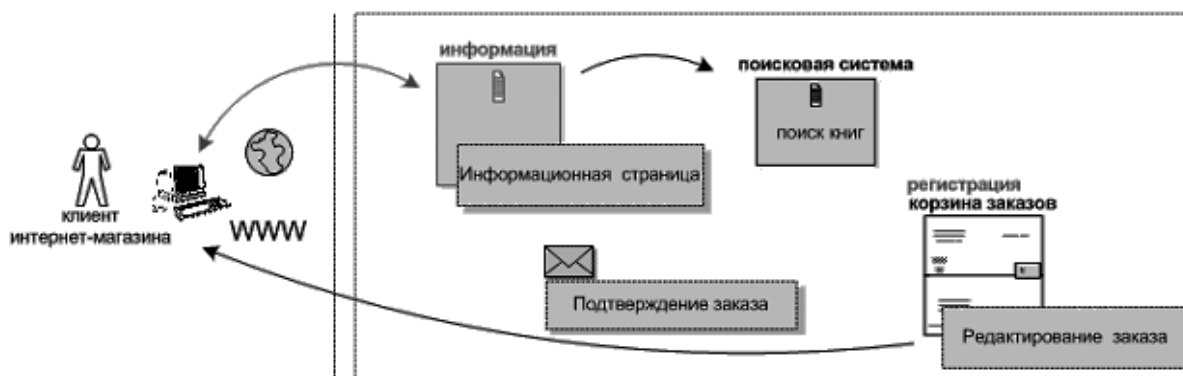


Рисунок 5.13 – Работа покупателя с интернет-магазином

При формировании заказа могут использоваться механизм электронного платежа для непосредственной оплаты суммы покупки с счета покупателя на счет продавца. В сети существует ряд платежных систем, которые обеспечивают безопасное использование электронных платежных карточек для оплаты покупок в Интернет. Диапазон других способов оплат достаточно велик: от дебетовых систем до систем расчетов на основе электронных денег.

Полученный заказ с минимальной задержкой, а, как правило, в реальном времени, становится доступен ответственному менеджеру. В его функции входит формирование заказов для их отправки заказчику (рис. 5.14). В большинстве случаев интернет-магазины для минимизации издержек не имеют собственного крупного склада, поэтому для обеспечения заказа формируется прямой дозаказ поставщику товара.



Рисунок 5.14 – Работа интернет-магазина по обработке и отправке заказа

К наиболее важным свойствам организации бизнеса в Интернете с точки зрения покупателя следует отнести следующие особенности.

Позитивные аспекты:

- независимость от времени, когда информационные ресурсы доступны каждый день в течение 24 часов на протяжении всего года;
- отсутствие привязки к местонахождению пользователя, что снимает ограничения на географическую удаленность покупателя;
- качество информационной поддержки подразумевает оперативный доступ ко всем информационным ресурсам продавца, ко всему спектру предлагаемых товаров и услуг.

Негативные аспекты:

- гарантии качества обслуживания ниже, чем при традиционной торговле. Это подразумевает возможные сбои при комплектации и доставке заказа;
- необходимость доступа к сети Интернет для поиска товара, формирования и управления заказом.

С точки зрения бизнеса организация интернет-торговли выглядит следующим образом.

Позитивные аспекты:

- отказ от торговых площадей, так как необходимость в дорогостоящих площадях отпадает, остается только склад;
- значительное сокращение штата, связанное с отсутствием торговых площадей;

- возможность витрины функционировать круглые сутки;
- предоставление информации на географически неограниченный регион.

Интернет обладает такими существенными преимуществами, как интерактивность, наличие обратной связи, возможность реализации индивидуального подхода к покупателю – персонализация.

### ***Chat-разговор с помощью сети IRC и электронной почты***

**IRC** (*Internet relay chat*) – это связка крупных сетей (Efnet, Dalnet, Undernet и др.), в каждой из которых сотни chat'ов и десятки тысяч пользователей. Официальный отчет истории IRC ведется с 1988 года. Именно тогда финский студент Джако, некоторое время, поговорив на многолинейных BBS, задался целью создать нечто похожее, но более глобального масштаба. Тогда и появилась первая сетка IRC-Efnet.

Сейчас любой уважающий себя сайт просто обязан иметь собственный сервис, позволяющий пользователям в интерактивном режиме общаться между собой.

### **Игры через Интернет**

Ни для кого уже не секрет, что игры занимают значительную часть жизни других людей. Играть можно против компьютера (интересно, но не очень), против одного противника (человека) с помощью модема и можно играть против многих противников с помощью локальных сетей или Интернета. Сейчас существует много серверов, которые предназначены исключительно для игр, таких как Quake, Quake II, Team Fortress, Warcraft II, Starcraft и др. Для того чтобы качество игры было приемлемым, необходимо обеспечить стабильную и высокоскоростную связь с Интернетом.

### **Списки рассылки**

Списки рассылки (mail-list) – это практически единственный сервис, не имеющий собственного протокола и программы клиента и работающий исключительно через электронную почту.

Идея работы списка рассылки состоит в том, что существует некий адрес электронной почты, который на самом деле является общим адресом многих людей – подписчиков этого списка рассылки. Вы посылаете письмо на этот адрес, например, на адрес u-lln@jet.msk.su (это адреса списка рассылки, посвященного обсуждению проблем локализации операционных систем класса UNIX), и ваше сообщение получают все люди, подписанные на этот список рассылки.

## **Перспективы развития Интернета. Интернет2**

Из-за начавшейся в середине 1990-х гг. активной коммерциализации Интернета пропускная способность его коммуникационных линий становится все более дефицитным ресурсом. Увеличиваются требования к пропускной способности каналов связи, обеспечению конфиденциальности доступных в Интернете и передаваемых информационных ресурсов, качеству сервисных услуг и т. п. В результате появился проект Интернет2.

Интернет2 – второе поколение Интернета, разрабатываемое и поддерживаемое исследовательским консорциумом Inernet2. Инициатива разработки Интернет2 принадлежит корпорации университетов США – USAID (University Corporation for Advanced Интернет Development). Разработчики Интернет2 сотрудничают с авторами других американских и зарубежных проектов (в том числе Канады и Мексики). В 2003 г. к созданию Интернет2 присоединилось большинство производителей телекоммуникационного оборудования. На начальном этапе появления и развития Интернет2 представлял собой большую сеть, связывающую вузы и исследовательские институты, с использованием входящих в нее высокоскоростных экспериментальных и частных сетей, а также специального программного обеспечения. В основу Интернет2 легла новая (шестая) версия протокола пакета передачи данных – Ipv6, разработанная международной организацией сообщества Интернета – IETF. Протокол Ipv6 должен заменить действующий протокол четвертой версии – Ipv4. Создана специальная организация, призванная способствовать продвижению этого протокола – IPv6 Forum. Технологии Интернет2 позволяют обеспечить скорость передачи данных до 10 Гбит/с и поддерживают средства Multicast (для одновременной широковещательной передачи данных нескольким абонентам сети), QoS (в том числе для обеспечения качества передач видео- и аудиоданных), а также использование высокоскоростных магистральных каналов. Пользователи Интернет2 могут одновременно оставаться пользователями обычного Интернета.

### **Организация семантической паутины**

Другая актуальная на сегодня концепция развития Всемирной паутины – создание семантической (осмысленной) паутины. Автор концепции семантической паутины также Тим Бернерс-Ли. Семантическая паутина (*semantic web*) – это надстройка над существующей Всемирной паутиной, которая призвана сделать размещенную в сети информацию более понятной для компьютеров.



В настоящее время компьютеры принимают довольно ограниченное участие в формировании и обработке информации в сети Интернет. Функции компьютеров в основном сводятся к хранению, отображению и поиску информации. В то же время создание информации, ее оценку, классификацию и актуализацию – все это по-прежнему выполняет человек. Как включить компьютер в эти процессы? Если компьютер пока нельзя научить понимать человеческий язык, то нужно использовать язык, который был бы понятен компьютеру. То есть в идеальном варианте вся информация в Интернете должна размещаться на двух языках: на человеческом языке для человека и на компьютерном языке для понимания компьютера. Семантическая паутина – это концепция сети, в которой каждый ресурс на человеческом языке был бы снабжен описанием, понятным компьютеру.

Программы смогут сами находить нужные ресурсы, обрабатывать информацию, классифицировать данные, выявлять логические связи, делать выводы и принимать решения. При широком распространении и грамотном внедрении семантическая паутина может вызвать революцию в Интернете.

### **Вопросы для повторения и самоконтроля**

1. Каково назначение межкомпьютерной связи? Ее основные способы.
2. Опишите технологию «клиент–сервер».
3. Что представляет из себя компьютерная сеть?
4. Каким образом преодолевается проблема несовместимости интерфейсов в компьютерных сетях?
5. Охарактеризуйте основные виды сетевых топологий.
6. Архитектура сети. Назовите характеристики распространенных сетевых архитектур.
7. Дайте краткую характеристику специального сетевого оборудования.
8. Что такое локальная сеть? В каких областях и с какой целью она применяется?
9. Назначение сети Интернет.
10. Как была создана сеть Интернет?
11. Кто такой Бернес Ли и какова его роль в создании Всемирной паутины?

12. Попробуйте приблизительно оценить размер Интернета и его российского фрагмента на текущий период.
13. Каким образом осуществляется подключение пользователя к сети Интернет?
14. Основные способы подключения пользователя к сети Интернет?
15. Каким образом осуществляется передача данных в сети Интернет?
16. Протоколы коммуникации TCP/IP.
17. Что такое IP-адрес?
18. Что такое доменная (DNS) система имен?
19. Сколько доменов имеет Россия? Как они называются?
20. Понятие универсального указателя ресурса URL.
21. Что такое сервисы Интернет? Перечислите основные сервисы сети Интернет.
22. Какие основные услуги предоставляет пользователям система WWW?
23. Что такое гипертекст, гипермедиа, WWW-страницы?
24. Назначение языка HTML и его последующих клонов.
25. Назначение браузеров.
26. Назначение электронной почты. Система адресации электронных адресов.
27. Как организованы системы информационного поиска сети Интернет? Какие поисковые документы существуют для этого?
28. Назначение каталогов ресурсов.
29. Поисковые машины. Какими параметрами определяется качество поиска?
30. Перечислите наиболее популярные поисковые системы в Интернете.
31. Назначение онлайн-энциклопедий и справочников.
32. Назначение программ пересылки файлов FTP.
33. Что такое web-сайт организации и методика его создания?
34. Интернет-торговля. Ее преимущества и недостатки.
35. Что такое Chat?
36. Что такое списки рассылок?
37. Интернет2. Почему возникла необходимость его создания?
38. Что такое семантическая паутина?

## Задания для самостоятельной работы

1. Первая глобальная компьютерная сеть носила имя:

- а) BITNet;
- б) ARPANet;
- в) NSFNet.

2. Провайдер Internet это:

- а) организация-поставщик услуг Internet;
- б) организация, занимающаяся созданием web-сайтов;
- в) периферийное устройство, служащее для связи с другим

компьютером.

3. FTP – это:

- а) почтовый клиент;
- б) программа IP-телефонии;
- в) протокол передачи файлов.

4. Задан URL-адрес web-страницы: <http://www.sgzt.com/sgzt/archive/content/2005/03/043>. Имя протокола доступа к этому информационному ресурсу:

- а) [sgzt/archive/content/2005/03/043](http://www.sgzt.com/sgzt/archive/content/2005/03/043);
- б) com;
- в) http;
- г) [www.sgzt.com](http://www.sgzt.com).

5. Доменная зона первого уровня, в которой может приобрести себе доменное имя юридическое лицо, зарегистрированное на территории РФ:

- а) .com;
- б) .ru;
- в) и в том и в другом.

6. Среди приведенных записей укажите корректный IP-адрес компьютера:

- а) 198.15.19.216;
- б) 298.15.19.216;
- в) 200,6,201,13;
- г) <http://www.ipc.ru>;
- д) [www.ip-address.com](http://www.ip-address.com).

7. Выберите корректный адрес для e-mail:

- а) Глеб@mur.ru;
- б) mur.ru@gleb;
- в) gleb@mur.ru;

- г) gleb.1@mur.mil;
- д) gleb @ mur. ru;
- е) gleb@mur;
- ж) <http://www.mur.ru/gleb>.

8. HTML – это:

- а) один из протоколов семейства TCP/IP;
- б) язык гипертекстовой разметки документа;
- в) язык программирования.

*Ответы на задания для самостоятельной работы*

- 1. Верные ответы: 2;
- 2. Верные ответы: 2;
- 3. Верные ответы: 3;
- 4. Верные ответы: 3;
- 5. Верные ответы: 3;
- 6. Верные ответы: 1;
- 7. Верные ответы: 3;
- 8. Верные ответы: 2.

## **Глава 6. ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ. КЛАССИФИКАЦИЯ КОМПЬЮТЕРОВ**

### **6.1. История развития средств обработки информации**

История счетных устройств насчитывает много веков.

Ниже в хронологическом порядке приводятся некоторые наиболее значимые события этой истории, их даты и имена участников<sup>7</sup>.

Около 30 000 до н. э. Пожалуй, самым древним из найденных инструментов считается кость с зарубками, найденная в древнем поселении Дольни Вестонци на юго-востоке Чехии в Моравии. Этот предмет получил название «вестоницкая кость».

Около 4000 до н. э. В египетских экономических текстах стали использовать символы цифр.

Около 3000 до н. э. В Древнем Шумере (Месопотамии) изобретен абак (простейшие счеты).

2112–1997 до н. э. В Месопотамии появилась первая позиционная шестидесятеричная система счисления.

Около 500 до н. э. Появились счеты, близкие к современным – с косточками на проволоке.

384–322 до н. э. Древнегреческий ученый и философ Аристотель разработал основы формальной логики, ввел понятие о переменных величинах, применил буквы для их обозначения.

310–280 до н. э. Древнегреческий математик Евклид в своей книге «Начала» изложил основы теории чисел.

III в. до н. э. В математике Месопотамии в состав цифр введен знак для нуля. В это же время майя использовали ноль в своей двадцатиричной системе счисления.

III в. н. э. Древнегреческий математик Диофант Александрийский разработал алгебраическую символику

III в. н. э. Римская непозиционная система счисления является самой распространенной непозиционной системой.

VII в. В Индии появилась десятичная позиционная система счисления с нулем, которая используется в настоящее время.

Начало 1500 годов. Леонардо да Винчи (1452–1519) создал эскизные наброски суммирующей вычислительной машины на зубчатых колесах, способной складывать 13-разрядные десятичные числа.

---

<sup>7</sup>Более подробно с историей развития вычислительной техники вы можете ознакомиться у Н. Колдовского [9], Б. Н. Малиновского [12].

1614 г. Шотландец Джон Непер (1550–1617) изобрел логарифмы. Вскоре после этого Р. Биссакар в 1564 году, а в 1657 году – независимо от него – С. Патридж создали логарифмическую линейку, основной счетный прибор инженера до середины XX века.

1623 г. Немецкий ученый Вильгельм Шиккард (1592–1636) разработал счетную машину для суммирования и умножения шестирядных десятичных чисел.

1641–1642 гг. Девятнадцатилетний французский ученый Блез создал действующую суммирующую машину («паскалину»).

1673 г. Немецкий ученый Вильгельм Готфрид Лейбниц (1646–1716), создал счетную машину для сложения и умножения двенадцатирядных десятичных.

1774 г. Сельский пастор Филипп Маттеос Хан разработал первую действующую счетную машину, имевшую коммерческий спрос.

1794 г. Во Франции был создан оптический телеграф Клода Шаппа. Это была первая надежная крупномасштабная сеть для передачи сообщений со стандартизированной системой кодирования.

1799 г. Во Франции Жозеф Мари Жакард (1752–1834) изобрел ткацкий станок, в котором для задания узора на ткани использовались перфокарты.

1821 г. Шарль-Ксавье Тома де Кольмар (1785–1870) создал первый механический калькулятор, который мог складывать, умножать, вычитать и делить.

1795 г. Математик Гаспар Прони (1755–1839) впервые в мире разработал технологическую схему вычислений, предполагающую разделение труда математиков на группы.

1830 г. Английский ученый Чарльз Беббидж (1791–1871) разработал проект аналитической машины. Только летом 2001 года машина Беббиджа была построена стараниями Дорона Суода, директора лондонского Музея науки.

1843 г. Программы вычислений на машине Беббиджа составлены дочерью Байрона Адой Августой Лавлейс (1815–1852). Их использовали для создания программ для первых ЭВМ.

1840–1860-е гг. Англичанин Джордж Буль (1815–1864) разработал алгебру логики (алгебру Буля). Соединил математическую логику с двоичной системой счисления и электрическими цепями американский ученый Клод Шеннон в 1936 г.

1846 г. Появился счислитель Петербургского учителя музыки Куммера, который серийно выпускался более 100 лет.

1855 г. Братья Джорж и Эдвард Шутц (George & Edvard Scheutz) из Стокгольма построили первый механический компьютер, используя работы Ч. Бэббиджа.

1867 г. Кристофер Шоулз (1819–1890) вместе со своим другом Карлом Глидденом изобрели пишущую машинку

1876 г. Английский инженер Александер Белл изобрел телефон.

1880 г. В.Т. Однер в России создал механический арифмометр с зубчатыми колесами, и в 1890 г. наладил его массовый выпуск. Под названием «Феликс» он выпускался до 50-х гг. XX века.

1890 г. Американский инженер Герман Холлерит создал статистический табулятор, использовавший 80-колонные перфокарты, которые использовались в первых трех поколениях компьютеров в качестве носителя информации.

В конце XIX века была изобретена перфоленга.

1892 г. Американский инженер Уильям Барроуз выпустил первый коммерческий сумматор.

1897 г. Английский физик Дж. Томсон сконструировал электронно-лучевую трубку.

1900–1901 гг. Итальянский физик Гульельмо Маркони и русский ученый Александр Попов изобрели радио.

1904–1906 гг. Сконструированы электронные лампы: диод и триод.

1919 г. Русский ученый Михаил Александрович Бонч-Бруевич (1888–1940) и английские ученые В. Икклз и Ф. Джордан (1919) независимо друг от друга создали электронное реле, названное англичанами триггером.

1919 г. Норвежский инженер Фредерик Розинг Бюль усовершенствовал устройство Холлерита и разработал принципы программируемого табулятора.

1923 г. Американский ученый русского происхождения В.К. Зворыкин изобрел иконоскоп – передающую электронную телевизионную трубку.

1930 г. Профессор Массачусетского технологического института (МТИ) Ванневер Буш (Vannevar Bush, 1890–1974) построил дифференциальный анализатор, с появлением которого связывают начало современной компьютерной эры.

1936 г. Английский математик Алан Тьюринг (1912–1954) и независимо от него Э. Пост выдвинули и разработали концепцию абстрактной вычислительной машины.

1938 г. Немецкий инженер Конрад Цузе (1910–1985), построил первый механический компьютер Z1. Создал первую в мире релейную вычислительную машину с программным управлением Z3 (1941 г.) и цифровую специализированную управляющую вычислительную машину Z4 (1943 г.).

1937 г. Американский физик болгарского происхождения Дж. В. Атанасов (John Atanasoff, 1903–1995) создал вместе со своим аспирантом Клиффордом Берри (Clifford Berry) работающую настольную модель ЭВМ. Это был первый в мире ламповый электронный цифровой компьютер ABC (Atanasoff Berry Computer).

В 1938 г. В телефонной компании Bell Laboratories создали первый двоичный сумматор.

1939 г. Джордж Стибитц (George Stibits) и Сэмюель Вильямс (Samuel Williams) создали Complex Number Calculator – калькулятор, складывающий комплексные числа, а также проводящий вычитание, умножение и деление. Он был первой машиной, к которой имелся удаленный доступ через телефонные линии в режиме разделенного времени.

1941 г. Инженер фирмы IBM Б. Фелпс в 1942 году создал экспериментальную модель электронного множительного устройства.

1942–1946 гг. Американский физик Джон Моучли (John Mauchly, 1907–1980) и Джон Эккерт (John Presper Eckert) создали вычислительную машину, которая стала известна под именем ENIAC (Electronic Numerical Integrator and Computer – электронный числовой интегратор и калькулятор)

1943 г. В Лондоне была построена машина Colossus на 1500 электронных лампах. Разработчики машины – М. Ньюмен и Т.Ф. Флауэрс.

1944 г. Под руководством американского математика Говарда Айкена (1900–1973) создана автоматическая вычислительная машина «Марк-1» с программным управлением. Она была построена на электромеханических реле.

1945 г. Вэннивер Буш впервые изложил идею создания гипертекста.

1945 г. Джон фон Нейман (1903–1957) сформулировал основные принципы работы и компоненты современных компьютеров. Под руководством Дж. Неймана в Принстонском институте перспективных исследований в 1952 г. была создана машина на электронных лампах МАНИАК (для расчетов по созданию водородной бомбы), а в 1954 г. еще одна, уже без его участия – «Джониак».



1946 г. Американский ученый-статистик Джон Тьюки предложил название БИТ (BIT – аббревиатура от BInary digiT).

1947–1948 гг. Группой при лаборатории Bell Telephone Laboratories, возглавленной Уильямом Брэдфордом Шокли (William Bredford Chockley, 1910–1989) изобретены транзисторы.

1947 г. Норберт Винер (Norbert Wiener 1894–1964) вводит в обращение термин «кибернетика».

1948 г. Под руководством академика С.А. Лебедева (1890–1974) и В.М. Глушкова разрабатываются отечественные ЭВМ: сначала МЭСМ – малая электронная счетная машина (1951 г., Киев), а затем в 1952 году в Москве – БЭСМ – быстродействующая электронная счетная машина.

1948 г. Ванг Ан (Wang An) изобрел запоминающее устройство на магнитных сердечниках, которое применялось в компьютерах до появления микросхем.

1948 г. Введен в действие первый в мире компьютер с хранимой программой Манчестерский Марк-1, созданный английскими учеными Том Килбурном (Tom Kilburn) и Фредди Вильямсом (Freddie Williams) из Манчестерского университета.

1948 г. Американский математик и инженер Клод Шеннон ввел понятие количества информации.

1949 г. В мае в Англии заработал EDSAC (Electronic Delay Storage Automatic Calculator, электронный автоматический вычислитель с памятью на линиях задержки) – первый действующий компьютер с хранимой программой – конструктор Морис Уилкис (Maurice Wilkes).

1949 г. М. Уилкс ввел систему мнемонических обозначений для машинных команд, названную языком ассемблера.

1949 г. Джон Моучли (John Mauchly) создал первый интерпретатор языка программирования под названием «Short Order Code».

1951 г. Под руководством Джона Моучли закончена работа по созданию первого коммерческого компьютера – UNIVAC (Universal Automatic Computer)

1951 г. Грейс Хоппер (Grace Hopper) разработала первую транслирующую программу, которую она назвала компилятором.

1951 г. Джей Форрестер запатентовал память на магнитных сердечниках.

1951 г. М. Уилкс представил доклад «Наилучший метод конструирования автоматической машины», который стал пионерской работой по основам микропрограммирования.

1951 г. М. Уилкс совместно с Д. Уиллером и С. Гиллом написали первый учебник по программированию «Составление программ для электронных счетных машин» (русский перевод – 1953 г.).

1952 г. В СССР в 1952–1953 гг. А.А. Ляпунов разработал операторный метод программирования (операторное программирование), а в 1953–1954 гг. Л.В. Канторович – концепцию крупноблочного программирования.

1952 г. В СССР в Москве была создана ЭВМ М-1.

1952 г. Фирма IBM выпустила свой первый промышленный компьютер IBM 701.

1952 г. Начало истории магнитной ленты как средства хранения компьютерных данных.

1953 г. В СССР выпущена первая серийная отечественная вычислительная машина «Стрела».

1954 г. Фирмой IBM разработан первый быстродействующий принтер, основанный на использовании вращающегося барабана со шрифтовым набором символов.

1955 г. В США изготовлен первый в мире компьютер на полупроводниковых транзисторах и диодах без использования электронных ламп – TRADIC (TRAnisitor DIgital Computer).

1955–1959 гг. Российские ученые А.А. Ляпунов, С.С. Камынин, Э.З. Любимский, А.П. Ершов, Л.Н. Королев, В.М. Курочкин, М.Р. Шура-Бура и другие создали программирующие программы – прообразы трансляторов. В.В. Мартынюк создал систему символьного кодирования – средство ускорения разработки и отладки программ. Заложены фундамент теории программирования (А.А. Ляпунов, Ю.И. Янов, А.А. Марков, Л.А. Калужин) и численных методов (В.М. Глушков, А.А. Самарский, А.Н. Тихонов). Моделируются схемы механизма мышления и процессов генетики, алгоритмы диагностики медицинских заболеваний (А.А. Ляпунов, Б.В. Гнеденко, Н.М. Амосов, А.Г. Ивахненко, В.А. Ковалевский и др.).

1955 г. Сотрудниками фирмы IBM под руководством Джона Бэкуса (John Backus) разработан первый алгоритмический язык FORTRAN.

1956 г. Фирмой IBM был разработан первый жесткий диск.

1956 г. Вернер Бухгольц (Werner Buchholz) в 1956 г. ввел в обращение термин «байт» (byte).

1957 г. В институте электронных управляющих машин была спроектирована ЭВМ для контроля космического пространства М-4.

1957 г. В Пензе под руководством Б.И. Рамеева создана одноадресная ламповая ЭВМ «Урал-1» общего назначения.

1958 г. Японская корпорация NEC разработала первый японский компьютер NEC-1101 и 1102.

1958 г. Bell Labs создала устройство (подобие модема) для передачи данных по телефонным линиям.

1958 г. Сеймором Крейем был разработан первый в мире суперкомпьютер, выполненный полностью на полупроводниковых элементах, CDC 1604.

1959 г. Дж. Маккарти и К. Стрейчи предложили концепцию разделения времени работы компьютера.

1959 г. Выпущена отечественная вычислительная машина Сетунь, работающая в троичной системе счисления.

1959 г. Создана машина М-20. На основе М-20 была создана уникальная многопроцессорная М-40.

1959 г. Начало выпуска в Минске ЭВМ «Минск-1».

1960 г. Роберт Нойс и Джек Килби независимо друг от друга создали монолитную интегральную схему.

1960 г. Разработан стандартизированный деловой язык программирования COBOL (Common business oriented language – общепринятый деловой ориентированный язык).

1960 г. Появился ALGOL (Algorithmic Language – алгоритмический язык).

1960 г. В СССР разработана первая полупроводниковая управляющая машина «Днепр» (В.М. Глушков, Б.Н. Малиновский).

1961 г. Американским профессором Джоном Маккартни разработан язык LISP (List processing language – язык обработки списков).

1961 г. Компанией DEC выпускается первый миникомпьютер PDP-1.

1961 г. Фирма IBM Deutschland реализовала подключение компьютера к телефонной линии с помощью модема.

1961 г. В Ереване начат серийный выпуск ЦВМ «Раздан-2».

1962 г. ЭВМ «Минск-2» выпущена г. в Минске.

1963 г. Утвержден американский стандартный код для обмена информацией – ASCII (American Standard Code Informatio Interchange).

1963 г. Фирма General Electric создала первую коммерческую СУБД (систему управления базами данных).

1963 г. В СССР под руководством В.М. Глушкова запускается в серийное производство компьютер «Промінь».

1963 г. Художник Харви Бэлл нарисовал первый смайлик.

1964 (по другим данным в 1965 г.) Сотрудник Стэнфордского исследовательского центра Дуглас Энгельбарт (Douglas Engelbart) продемонстрировал работу первой компьютерной мыши.

1964 г. Фирма IBM запустило в производство семейство ЭВМ IBM 360 (System 360), ставших первыми компьютерами третьего поколения.

1964 г. Профессорами Дартмутского колледжа Томом Куртцем (Tom Kurtz) и Джоном Кемени (John Kemeny) был разработан язык BASIC (Beginners all-parpouse symbolic instraction code – многоцелевой язык символических инструкций для начинающих).

1964 г. Созданы ЭВМ М4-2М и М4-3М.

1964 г. Начало выпуска ряда ЭВМ Урал: Урал-11, -14, -16.

1965 г. Сеймур Пейперт (Seymour Papert) разработал язык LOGO – компьютерный язык для детей.

1965 г. Ларри Робертс впервые организовал взаимодействие между компьютерами на базе коммутации пакетов.

1965 (1967) гг. Разные источники называют дату с разбросом в 2 года. Под руководством С.А. Лебедева организован крупносерийный выпуск БЭСМ–6, – самой быстродействующей на то время ЭВМ в мире.

1965 г. IBM совместно с группой пользователей SHARE – разработали язык программирования PL/1 (Programming language – универсальный программно-ориентированный).

1965 г. Разработана ЭВМ «Минск-22» и «Минск-32».

1965 г. Е Ереванском научно-исследовательском институте математических машин разработаны ЭВМ семейства «Наири».

1965 г. В Киеве создана машина «МИР-1».

1966 г. ЭВМ «Раздан-3» серийно выпускается с 1966.

1967 г. Компания IBM начала производить 8 дюймовую дискету для компьютера, изобретенную Йосиро Накамацу.

1968 г. В 1968–1970 гг. профессор Никлаус Вирт создал язык PASCAL.

1968 г. Голландский ученый Эдсгер Дейкстра разработал концепцию структурного программирования.

1968 г. Компания BBN (Bolt, Beranek and Newman, Inc.) начала работу по постройке сети ARPANET.

1968 г. Основана фирма Intel.

1969 г. Фирма IBM разделила понятия аппаратных средств (hardware) и программные средства (software).

1969 г. Профессор математики МТИ (Массачусетского технологического института) Сеймур Пейперт и его коллеги создали новый язык на основе Лиспа, назвав его Лого, «слово».

1969 г. Впервые в ЭВМ МИР-2 был применен дисплей со световым пером.

1969 г. Разработана ЭВМ «РУТА-110».

1969 г. В США началось создание оборонной компьютерной сети – прародителя современной всемирной сети Internet. 29 октября принято считать днем рождения Сети. В этот день была предпринята самая первая, попытка дистанционного подключения к компьютеру.

1970 г. Сотрудник Национальной радиоастрономической обсерватории Чарльз Мурр создал язык программирования ФОРТ.

1970 г. Математик Эдгар Кодд (Edgar Codd), сотрудник IBM, описал концепцию реляционных баз данных и сформулировал 12 правил Кодда.

1971 г. Первым в истории адресом электронной почты стал tomlinson@bbn-tenexa.

1971 г. Фирмой Intel (США) создан первый микропроцессор Intel-4004, содержащий все основные компоненты центрального процессора.

1971 г. Появился компьютер IBM/370 модель 145 – первый компьютер, в основной памяти которого использовались исключительно интегральные схемы.

1971 г. В свет выходит первый в мире карманный калькулятор Rocketronic.

1971 г. Нолан Башнелл сделал игру Pong («Вонь»), ставшую первой настоящей и популярной компьютерной игрой.

1971 г. Французский ученый Алан Колмари разработал язык логического программирования Пролог (PROgramming in LOGic).

1972 г. Сеймур Крей организовал фирму Cray Research.

1972 г. Созданы шесть моделей компьютеров Единой системы (ЕС ЭВМ).

1972 г. Деннис Ритчи из Bell Lab's разработал язык программирования «С» (Си).

1972 г. Кен Томпсон и Деннис Ритчи разработали операционную систему UNIX.

1972 г. Концепция виртуальной машины была впервые реализована на компьютерах семейства IBM/370.

1972 г. Появился протокол Telnet.

1973 г. Рождение Ethernet произошло стараниями Роберта Меткалфа в лаборатории Xerox PARC.

1973 г. Разработан учеными университета Люммини во Франции под руководством Колмероз язык PROLOG (Programmation en logique – логическое программирование).

1973 г. Фирмой IBM была впервые разработана память на жестких дисках типа «винчестер».

1973 г. Выпускается компьютер Micral, в описании которого впервые используется термин «микрокомпьютер».

1974 г. Фирма Intel разработала первый универсальный восьмиразрядный микропроцессор.

1974 г. Эдвард Робертс, молодой офицер ВВС США, инженер-электронщик фирмы MITS, создал первый персональный компьютер Altair.

1974 г. Первый компьютер неудачно сыграл в шахматы с человеком: шахматист легко выиграл.

1974 г. Internet Network Working Group (INWG), руководимая Винтоном Серфом (Vinton Cerf, Stanford Research Institute) разработала универсальный протокол передачи данных и объединения сетей Transmission Control Protocol/Internet Protocol (TCP/IP) – сердце Internet.

1974 г. Начало выпуска моделей второго ряда ЕС ЭВМ.

1975 г. Гарри Килдалл из фирмы Digital Reseach разработал операционную систему CP/M.

1975 г. Фирма IBM представила портативный мини-компьютер IBM 5100 Portable Computer.

1975 г. Студенты Гарвардского университета Билл Гейтс и Пол Аллен написали программное обеспечение для персонального компьютера Altair на основе языка Бейсик.

1975 г. Компания Xerox выпускает первый миникомпьютер, который комплектовался мышкой и имел графический интерфейс.

1975 г. Фирма IBM начала продажу лазерных принтеров.

1976 г. Стив Возняк из компании Hewlett-Packard создал принципиально новый микрокомпьютер Apple-I, а затем основали компанию Apple.

1976 г. В марте фирма Cray Research выпустила суперкомпьютер Cray-1.

1976 г. Появилась дискета диаметром 5,25 дюйма.

1976 г. Зарегистрирована торговая марка «Microsoft».

1976 г. Майкл Шрайер (Michael Shraye) завершает работу над первым текстовым процессором Electric Pencil («Электронный карандаш») для микрокомпьютеров.

1976 г. Питер Нортон (Peter Norton) создал комплект ставших знаменитыми на весь мир полезных утилит – Norton Utilities.

1977 г. Были запущены в массовое производство три персональных компьютера: Apple-2, TRS-80 и PET.

1977 г. Разработан мини-компьютер VAX-11/780 первый 32-разрядный представитель нового семейства фирмы DEC.

1978 г. Фирма Intel выпустила микропроцессор.

1978 г. Epson анонсирует матричный принтер MX-80.

1978 г. Протокол управления передачей данных (Transmission Control Protocol - TCP) разделен на TCP и Internet Protocol (IP). Таким образом, возник хорошо известный протокол TCP/IP.

1978 г. В Чикаго заработала первая BBS – электронная доска объявлений.

1979 г. Фирма Intel выпустила микропроцессор 8088.

1979 г. Дэн Бриклин и Боб Фрэнкстон создали программу VisiCalc – первую в мире электронную таблицу.

1979 г. Англичанин Клайф Синклер создал первый домашний компьютер ZX80, ставший предшественником для ZX81 и невероятно популярного ZX Spectrum.

1979 г. Microsoft представила Basic – первый интерпретатор языка высокого уровня, предназначенный для 16-разрядных машин на базе процессора 8086.

1979 г. Коллектив под руководством М.А. Карцева разработал многопроцессорную систему М-10.

1979 г. В стенах Института проблем управления АН СССР завершается разработка высокопроизводительной вычислительной системы ПС-2000.

1979 г. Motorola анонсирует 16-разрядный процессор MC68000.

1979 г. Появился оптический компакт-диск (CD), разработанный фирмы Philips и предназначенный для прослушивания музыкальных записей.

1979 г. В Японии и США были разработаны первые сотовые телефоны.

1979 г. Вейн Ратлифф разработал базу данных Vulcan (Ashton-Tate впоследствии развернула ее продажи под именем dBase II).

1980 г. Hewlett-Packard представляет интегрированную разработку – микрокомпьютер HP-85.

1980 г. Появился язык ADA, названный в память об Аде Лавлейс – первой программистке в истории вычислительной техники.

1980 г. Японские компании Sharp, Sanyo, Panasonic, Casio и американская фирма Tandy вынесли на рынок первый карманный компьютер, обладающий всеми основными свойствами больших компьютеров.

1981 г. Фирма Compaq выпустила первый Laptop.

1981 г. Никлаус Вирт разработал язык программирования МОДУЛА-2.

1981 г. Создан первый портативный компьютер – Osborne 1.

1981 г. Компания IBM представляет свой компьютер, который в дальнейшем положил начало понятию «IBM-совместимый компьютер» – IBM 5150 (IBM PC). Для IBM PC фирма Microsoft разработала операционную систему MS-DOS.

1981 г. Появились дискеты диаметром 3,5 дюйма, выпущенные корпорацией Sony. А с 2007 года они сняты с производства.

1981 г. Дэвид Брэдли встроил в клавиатурный код команду для «горячей» перезагрузки и придумал <Ctrl>+<Alt>+<Del>.

1981 г. Начало выпуска в Киеве управляющего вычислительного комплекса (УВК) СМ-1420.

1982 г. Создан 16-разрядный процессор Intel 80286.

1982 г. Фирма Sun начала выпускать первые рабочие станции.

1982 г. Английской фирмой Inmos был создан язык ОККАМ.

1982 г. Питер Нортон написал программу Unerase.

1982 г. Появление первой версии AutoCAD и языка PostScript.

1982 г. Commodore Business Machines анонсирует Commodore 64.

1983 г. Корпорация Lotus Development начала продажи электронной таблицы Lotus 1-2-3 версии 1.0 под MS-DOS. Автор разработки Митчелл Кэпор (Mitchell David Kapor).

1983 г. Свою первую мышь Bus Mouse для IBM PC выпустила фирма Microsoft, а тремя годами позже появилась другая – InPort Mouse.

1983 г. Компания Visicorp выпускает Visi On – интегрированную графическую операционную среду для приложений на персональных компьютерах.



1983 г. Microsoft анонсировала на рынок Microsoft Windows как графическое приложение для операционной системы MS-DOS.

1983 г. Компания Borland выпустила Turbo Pascal 1.0, ставший прародителем Delphi, и знаменитый компилятор Андерса Хейльсберга (Anders Hejlsberg), с которого начала отсчет новая эра интерактивных систем программирования.

1983 г. Atari Inc. представляет домашний компьютер Atari 1200XL.

1983 г. Корпорация Apple Computers построила персональный компьютер Lisa – первый офисный компьютер, управляемый манипулятором мышью.

1983 г. Фирма IBM выпускает IBM 5160, получившая второе название PC XT.

1984 г. Вышла в свет MS-DOS 3.1 – первая версия MS-DOS, поддерживающая локальную сеть.

1984 г. 22 января принято считать днем рождения компьютеров марки Macintosh.

1984 г. Sony и Philips разрабатывают стандарт CD-ROM. Также разработаны стандарты MIDI и DNS.

1984 г. Французский математик Филипп Кан основал компанию Borland International и стал продавать компилятор Turbo Pascal для персональных компьютеров.

1984 г. Hewlett-Packard выпустила первый принтер серии LaserJet.

1984 г. Появилась некоммерческая компьютерная сеть FIDO. Ее создатели – Том Дженнингс и Джон Мэдил. В 1995 году в мире насчитывалось около 20 тысяч узлов этой сети, объединяющих 3 млн. человек.

1984 г. Разработан «Агат» – советский универсальный 8-разрядный персональный компьютер.

1985 г. Сеймур Крей создал суперкомпьютер CRAY-2.

1985 г. Летом фирма IBM закрыла свою последнюю фабрику по выпуску перфокарт.

1985 г. Появился первый русский текстовый процессор Лексикон.

1985 г. Выпущен бытовой компьютер «Электроника БК0010-01» (ДВК-1), а затем и ДВК-2.

1985 г. В мае компания Microsoft официально объявила о создании системы электронных таблиц Excel (как для Macintosh, так и для IBM PC).

1985 г. Появился новый язык программирования C++. Автор Бьерн Страуструп.

1985 г. 17 октября корпорация Intel открыла линейку 386DX выпуском 16-МГц процессора.

1985 г. Начались продажи первой версии графической операционной среды Windows 1.0.

1985 г. Была основана компания ATI – разработчик и изготовитель видеоплат.

1986 г. На клавиатуре впервые появляются клавиши управления курсором и отдельный блок с цифровыми клавишами.

1986 г. Под эгидой IAB (Internet Activities Board – Координационный совет Интернета) образуются технические группы IETF (Internet Engineering Task Force – Рабочая группа по развитию Интернета) и IRTF (Internet Research Task Force – Рабочая группа по исследованию Интернета).

1986 г. Создан алгоритмический язык Turbo Pascal 3.02.

1987 г. IBM заявляет о выпуске компьютеров нового поколения PS/2, оснащенных операционной системой OS/2.

1987 г. ATI представляет на рынок EGA Wonder and VGA.

1987 г. Microsoft представляет первую реализацию электронной таблицы под Windows – Microsoft Excel 2.0.

1988 г. Даниел Хиллис, главный конструктор фирмы Thinking Machines, построил суперкомпьютер Connection Machine.

1988 г. Фирма Apple разработала систему Hyper Card.

1988 г. Появилась Windows 2.1, позже переименованная в Windows/286.

1988 г. Выход в свет антивирусной программы Д.Н. Лозинского Aidstest.

1988 г. Создание IANA (Internet Assigned Numbers Authority) – Агентства по выделению имен и уникальных параметров протоколов Интернета, полномочного органа Общества Интернет (Internet Society, ISOC) и Федерального Совета Сети (Federal Networking Council, FNC).

1988 г. IBM представила семейство серверов среднего класса AS/400.

1988 г. Motorola представила семейство RISC-микропроцессоров 88000.

1988 г. Роберт Тарран Моррис (младший) запустил в Internet червя, нацеленного на маршрутизаторы, который вывел из строя от 10 до 20 процентов компьютеров, подключенных тогда в Internet. По-

сле этого был образован координационный центр «скорой компьютерной помощи» – Computer Emergency Response Team (CERT).

1988 г. Медицинский журнал American Journal of Industrial Medicine опубликовал данные о том, что у женщин, работающих в период беременности за дисплеями более 2 часов в сутки, на 80% повышается риск неблагоприятного течения беременности.

1988 г. Мультипликационный фильм Tin Toy компании Pixar Animation Studios, созданный методом компьютерной анимации, стал первым произведением этого жанра, удостоенным «Оскара» в номинации за лучший короткометражный анимационный фильм.

1989 г. Фирма Intel выпустила микропроцессор Intel 486 DX.

1989 г. Creative Labs выпускают звуковую карту Sound Blaster, название которой впоследствии станет нарицательным.

1989 г. Microsoft выпустила текстовый процессор WORD.

1989 г. Разработан формат графических файлов GIF.

1989 г. Корпорация Microsoft выпустила графическую оболочку MS Windows 3.0.

1990 г. Выпуск и ввод в эксплуатацию векторно-конвейерной суперЭВМ «Эльбрус 3.1».

1990 г. В базе данных InterNIC официально зарегистрирован домен первого уровня .SU (Soviet Union – Советский Союз).

1990 г. Тим Бернерс-Ли (Tim Bernes-Lee) разработал язык HTML (HyperText Markup Language – язык разметки гипертекста; основной формат документов) и прототип Всемирной паутины, а также редактор Web-страниц и коммуникационные программы – и все это в операционной системе NextStep на рабочей станции Next. 17 мая 1991 г. состоялось первое подключение первого Web-сервера.

1990 г. Cray выпустил суперкомпьютер Cray Y-MP C90.

1990 г. Состоялся официальный выход графической программы Photoshop 1.0. созданного Томасом Кноллом (Thomas Knoll).

1990 г. Выпущен первый процессор, разработанный специально под архитектуру портативных компьютеров, – Intel 386SL.

1990 г. IBM представила семейство рабочих станций RS/6000 на базе RISC-процессоров.

1990 г. Toshiba выпустила SPARC LT – ноутбук на базе RISC-процессора.

1990 г. Компания Motorola начала выпускать микропроцессоры 68040.

1990 г. Приступила к работе компания The World – первый в мире коммерческий провайдер коммутируемого доступа в Internet.

1990 г. Последний год в истории ЕС ЭВМ – на развитие этого семейства было отпущено всего 100 млн рублей.

1990 г. Начала действовать общедоступная компьютерная сеть Relcom (RELIable COMMunications).

1991–1992 гг. Финский студент Линус Торвалдс (Linus Torvalds) распространил среди пользователей Интернета первый прототип своей операционной системы Linux.

1991 г. Появилось новое творение Apple – Macintosh Classic.

1991 г. Apple Computer выпустила первое семейство портативных компьютеров PowerBook.

1991 г. Состоялся дебют языка Microsoft Visual Basic 1.0 for Windows (VB).

1991 г. Появился логотип Intel Inside®.

1991 г. Разработан процессор Intel 486SX.

1991 г. Apple Computer начинает продажи QuickTime 1.0 (кодовое имя Warhol), первоначально разработанной в качестве мультимедийного дополнения к Mac OS 7.

1991 г. Hewlett-Packard представила карманный компьютер 95LX весом 310 г.

1991 г. Sun Microsystems разрабатывает язык программирования Oak, который в дальнейшем превратился в Java.

1991 г. Изготовлено четыре экземпляра «Электроники СС БИС» – последней ЭВМ, самостоятельно созданной в СССР.

1992 г. DEC представил первый 64-битный процессор RISC Alpha.

1992 г. Компания IBM представила семейство портативных компьютеров ThinkPad.

1992 г. Компания Sun Microsystems предложила систему SPARCstation 10 с возможностью многопроцессорной обработки.

1992 г. Корпорация Microsoft начала поставки базы данных Microsoft Access Database для Windows на основе бета-версии Windows NT и Windows 3.1.

1992 г. Стив Дириг и Стефен Каснер разработали систему многоадресной рассылки на базе IP для передачи аудио и видеопакетов по Internet.

1992 г. Сформулирован устав сообщества Internet.

1992 г. Джефф Хокинс и Донна Дубински основали компанию Palm Computing для разработки карманных устройств PDA. Первым продуктом Palm стал flop Zoomer.

1992 г. Компьютерный лексикон обогатился выражением *surfing the Internet*, что на жаргоне российских компьютерщиков звучит как «ходить по Internet».

1993 г. Появился первый web-браузер Mosaic, разработанный Эриком Бина и Марком Андресеном. На NCSA Mosaic основан Internet Explorer.

1993 г. Фирма Intel выпустила 64-разрядный микропроцессор Pentium.

1993 г. Появился формат сжатия видео MPEG.

1993 г. Появилась операционная система Windows NT (New Technology).

1993 г. Джерри Янг (Jerry Yang) и Дэвид Фило (David Filo), выпускники Стэнфордского университета создали Интернет-путеводитель Yahoo!

1993 г. Apple Computer выпускает Newton MessagePad – первое устройство клавиатурных карманных компьютеров.

1993 г. Трафик World Wide Web вырос за год на 341634 %.

1993 г. Учреждается заядлый враг ATI – NVIDIA

1993 г. Вышел в свет российский программный продукт оптического распознавания FineReader компании BIT Software (ныне АБВУУ).

1993 г. Компания ЭЛВИС+ (Электронные вычислительные и информационные системы) создает первый в Рунете общедоступный Web-сайт.

1994 г. Начало выпуска фирмой Apple Computers серии – Power PC.

1994 г. Компания Netscape Communication выпустила браузер Netscape Navigator 1.

1994 г. Появились первые рекламные баннеры.

1994 г. Iomega представляет свой дисковод Zip и соответствующие диски емкостью 25 и 100 Мбайт.

1994 г. Представлена полнофункциональная операционная система Linux 1.0.

1994 г. Джефф Безос разрабатывает идею интерактивного книжного магазина Amazon.com, а по существу – стандарт электронной коммерции.

1994 г. Стивен Кирш запускает InfoSeek – механизм поиска в Internet.

1994 г. Web теснит Telnet и становится второй по популярности службой в Internet.

1994 г. Зарегистрирован новый домен .ru, который начал постепенно теснить домен .su, существовавший с сентября 1990-го.

1995 г. Фирма Microsoft выпустила в свет операционную систему Windows 95.

1995 г. Фирма Microsoft выпустила браузер Internet Explorer.

1995 г. Фирма Intel выпустила микропроцессор Pentium Pro.

1995 г. Фирма DEC объявила о выпуске пяти новых моделей персональных компьютеров Celebris XL.

1995 г. Представлен первый клон компьютера Macintosh.

1995 г. Advanced Micro Devices (AMD) представила микропроцессор K5.

1995 г. В мае группа специалистов из Sun Microsystems, возглавляемая Джеймсом Гослингом, представила платформенно независимый язык программирования Java.

1995 г. IBM признала, что существует возможность сбоя при смене дат – проблема 2000 года – и обнародовала план предоставления своим клиентам услуг, инструментария и средств поддержки для обеспечения совместимости с 2000 годом.

1995 г. Кинокомпания Walt Disney представила первый полнометражный мультипликационный фильм, полностью реализованный компьютерными средствами, Toy Story (на российском видеорынке – «История игрушек»).

1996 г. Компания Palm представляет карманные компьютеры Pilot 1000 и Pilot 5000.

1996 г. Фирма Microsoft выпустила Internet Explorer 3.0.

1996 г. Фирма Microsoft выпустила операционную систему для карманных компьютеров Windows CE 1.0.

1996 г. Появление технологии Intel Video Phone в составе видеоконспекта Intel Create & Share Camera Pack позволило пользователям сети Интернет полноценно общаться с человеком в любой точке земного шара.

1996 г. Фирма «Партия» открывает первое в России бесплатное интернет-кафе.

1996 г. Российская антивирусная компания «ДиалогНаука» впервые в Глобальной сети (не только в Рунете) предложила онлайн-новую проверку на вирусы.

1996 г. Открыта поисковая система «Рамблер».

1996 г. Корпорация Sun Microsystems представила 64-разрядное семейство рабочих станций Ultra.

1996 г. Компания Palm Computing выпускает карманный компьютер Pilot PDA, получивший название PalmPilot.

1996 г. Корпорация Microsoft представила технологию ActiveX – набор средств разработки Internet-приложений.

1996 г. Впервые вышла в эфир MSNBC – кабельная сеть круглосуточной службы новостей, созданная Microsoft и NBC.

1996 г. CompTek представила программные продукты серии Yandex, предназначенные для полнотекстового поиска с учетом морфологии русского языка.

1996 г. «1С» создает первую в российском компьютерном бизнесе франчайзинговую сеть.

1996 г. Появилась технология записи CD-RW, разработанная промышленным консорциумом, в который вошли компании Ricoh, Philips, Sony, Yamaha, Hewlett-Packard и Mitsubishi Chemical Corporation

1997 г. Число пользователей Интернета в мире превышает 100 миллионов.

1997 г. Компания Sun Microsystems приняла стандарт объектно-ориентированного языка программирования Java.

1997 г. Фирма Apple выпустила операционную систему Macintosh OS 8.

1997 г. Появился первый процессор из серии Pentium MMX Mobile, предназначенный для мобильных устройств и ноутбуков.

1997 г. Корпорация Microsoft представила операционную систему для мобильных устройств Windows CE 2.0 (кодовое название Mercury).

1997 г. Гарри Каспаров со счетом 3,5:2,5 проиграл компьютерной программе Deep Blue.

1997 г. Intel выпускает первые процессоры с технологией MMX. Чуть позже появляются процессоры Pentium II.

1997 г. Началось разделение компьютеров на разные платформы (под Intel или AMD).

1997 г. «Лаборатория Касперского» стала самостоятельной компанией.

1998 г. Корпорация Microsoft выпускает операционную систему Windows 98.

1998 г. Появление стандарта беспроводной передачи голоса и данных Bluetooth, разработанного компанией Ericsson.

1998 г. Открылась бесплатная почтовая служба Mail.ru <http://www.mail.ru>.

1998 г. AMD выпускает процессор K6-300 МГц.

1998 г. Intel выпускает процессоры Pentium II на новом ядре Deschutes.

1999 г. Корпорация Intel представляет процессор Intel® Pentium® III.

1999 г. Intel начинает выпускать бюджетный процессор Celeron.

1999 г. Количество пользователей Интернета в мире превышает 201 миллион.

2000 г. Выход в свет операционной системы Windows 2000.

2000 г. Корпорация Intel представляет процессор Pentium® 4.

2000 г. IBM создала новый суперкомпьютер ASCI White.

2000 г. Персональным компьютерам исполняется 20 лет. В период с 1981 по 2000 год продано 835 миллионов компьютеров.

2000 г. Компания Apple выпускает персональный музыкальный проигрыватель iPod.

2000 г. Число пользователей Интернета в мире превышает 400 миллионов.

2000 г. С ноября 2000 года официальным названием главного продукта «Лаборатории Касперского» стал Антивирус Касперского®, который уже защищал не только офисные компьютеры и компьютеры рядовых пользователей, но также осуществлял защиту рабочих станций, файловых и почтовых серверов под управлением Windows, Linux и FreeBSD, сетей Novell NetWare, межсетевых экранов.

2001 г. Фирма Intel анонсирует процессор Itanium.

2001 г. Intel выпускает процессор Pentium 4.

2001 г. AMD выпустила процессор Athlon XP.

2002 г. Выпуск в свет операционной системы Windows XP и офиса соответствующего имени.

2003 г. Корпорация Intel представляет технологию Intel® Centrino® для мобильных ПК.

2003 г. Выпущен доработанный Office 2003.

2004 г. Впервые в праздничный сезон 2004 года объем продаж ноутбуков превысил объем продаж телевизоров.

2006 г. Корпорация Intel выпускает технологию Intel® ViiV™, торговую марку домашних ПК для отдыха.

2006 г. Корпорация Intel объявляет о выпуске технологии Intel® vPro™, новой платформе ПК, оптимизированной для предприятий

2006 г. Корпорация Intel выпускает процессор Intel® Core™2 Duo.



2006 г. Выпуск в свет операционной системы Alta Vista и промышленный выпуск двухядерных процессоров Intel для персональных компьютеров.

2006 г. По оценкам корпорации Intel, во всем мире к Интернету подключено около 1,3 миллиарда компьютеров.

2010 г. .рф – национальный домен верхнего уровня для России. Введен 12 мая 2010 года. Первый в Интернете домен на кириллице. Отличием от введенного ранее домена «.ru» является то, что в домене «.рф» все имена второго уровня пишутся исключительно кириллицей.

2012 г. Яндекс.браузер впервые был представлен 1 октября 2012 года на технологической конференции Yet another Conference. Яндекс.браузер (browser.yandex.com), создан компанией «Яндекс» на основе движка Blink, используемого в открытом браузере Chromium.

Современная история делается на ваших глазах, и вы сами участвуете в ее создании.

## **6.2. Этапы развития вычислительной техники. Поколения электронных вычислительных машин**

Этапы развития вычислительной техники принято называть *поколениями*. Этапы развития вычислительной техники и признаки, отличающие одно поколение от другого, приведены в табл. 6.1.

Таблица 6.1 – Этапы развития вычислительной техники

Этап	Признак, отличающий одно поколение от другого
1. Ручной – с 50-го тыс. до н. э. 2. Механический – с середины XVII в. 3. Электромеханический – с 90-х гг. XIX в. 4. Электронный – с 30-х гг. XX в.	1. Элементная база. 2. Быстродействие. 3. Объем оперативной памяти. 4. Устройства ввода-вывода. 5. Программное обеспечение

Краткая характеристика этапов развития ЭВМ приведена в таблицах 6.2. и 6.3.

Таблица 6.2 – Характеристика этапов развития ЭВМ

Поколение	Элементная база	Быстродействие (операций в секунду) <hr/> Объем ОП	Устройства ввода-вывода	Програмное обеспечение	Пример
Первое поколение, после 1946 г.	Электронные лампы, реле	$3 \times 10^5$ <hr/> 64 Кб	Пульт управления, перфокарта	Машинные языки	ENIAC, MARK-3, SWAC, IAS, BINAC, UNIVAC, MANIAC, WhirlWind-1, ORDVAC, IBM 701 (США) Gamma-40 (Франция) LEO, DEDUCE (Англия) МЭСМ, БЭСМ, Минск-1, Урал-2, М-20 (СССР)
Второе поколение, после 1955 г.	Транзисторы	$3 \times 10^6$ <hr/> 512 Кб	Перфокарты, перфоленты, АЦПУ, магнитный барабан	Алгоритмические языки, диспетчерские системы, пакетный режим	IBM 701, RCA-501, IBM 7090, LARC, Stretch (США) ATLAS (Англия) Раздан, Наири, Минск, МИР, Урал, Днепр, М-400, БЭСМ-6, БЭСМ-4, Минск-22, Минск-32 (СССР)
Третье поколение, после 1964 года	Интегральные схемы (ИС)	$3 \times 10^7$ <hr/> 16 МГб	Видеотерминальные системы	Операционные системы, режим разделения времени	PDP-8, PDP-11, V3500, IBM 360 (США) ЕС ЭВМ, СМ ЭВМ (СССР)
Четвертое поколение, после 1975 г.	Большие интегральные схемы (БИС) и сверхбольшие интегральные схемы (СБИС)	более $3 \times 10^7$ <hr/> более 16 МГб	Цветной графический дисплей, мышь	Базы и банки данных	ILLIAS 4, Cray-серией, Burroghs (США) ЕС 1191, ЕС 1766, Эльбрус (СССР)
Пятое поколение, после 1982 г.			Оптические, лазерные устройства, устройство голосовой связи	Экспертные системы	

Таблица 6.3 – Поколения ЭВМ

Параметр ЭВМ	Первое 1946–1955 гг.	Второе 1955–1965 гг.	Третье		Четвертое после 1980 г.
			1965–1970 гг.	после 1970 г.	
Основной элемент	Реле, электронные лампы	Транзисторы	Интегральная схема (ИС)	Большая интегральная схема (БИС)	Сверхбольшая интегральная схема (СБИС)
Быстродействие	1 мс	1 мкс	10 нс	1 нс	< 1 нс
Плотность упаковки, элементов/см <sup>3</sup>	0,1	2–3	10–20	1000	> 10000

После создания в 1946 г. в США модели ENIAC (рис. 6.1) был дан мощный импульс развитию универсальных ЭВМ, стимулировавший появление в ряде стран моделей ЭВМ, составивших первое поколение. На протяжении более 60 лет развития вычислительной техники появилось, сменяя друг друга, несколько поколений ЭВМ.

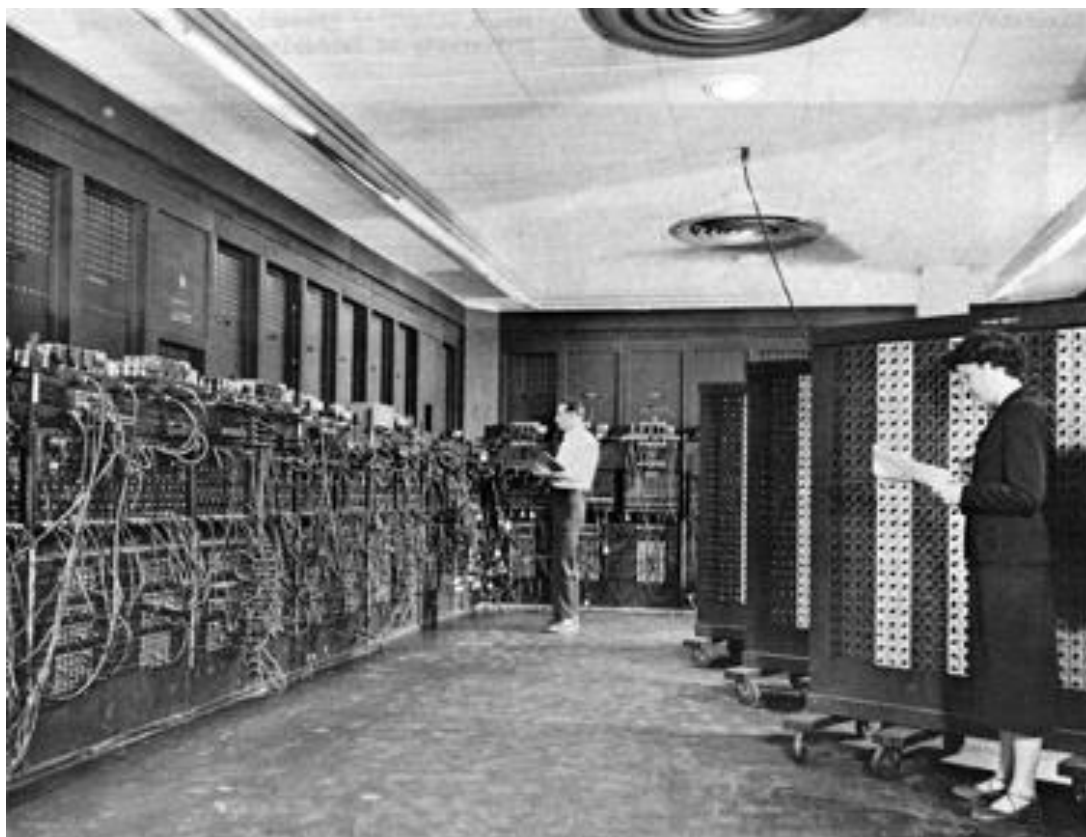


Рисунок 6.1 – ЭВМ первого поколения ENIAC

### *Первое поколение*

ЭВМ первого поколения в качестве элементной базы использовали электронные лампы и реле; оперативная память выполнялась на триггерах, позднее на ферритовых сердечниках; быстродействие было, как правило, в пределах 5–30 тыс. арифметических оп/с. Компьютеры отличались невысокой надежностью, требовали систем охлаждения и имели значительные габариты. Процесс программирования требовал значительного искусства, хорошего знания архитектуры ЭВМ и ее программных возможностей. На первых порах данного этапа использовалось программирование в кодах ЭВМ (машинный код), затем появились автокоды и ассемблеры.

Первой действующей ЭВМ 1-го поколения стал ENIAC (США, 1945–1946 гг.). Его название по первым буквам соответствующих английских слов означает «электронно-числовой интегратор и вычислитель».

Первая отечественная ЭВМ 1-го поколения – МЭСМ (Малая электронно-счетная машина) – была создана в 1951 г. под руководством С.А. Лебедева, крупнейшего советского конструктора вычислительной техники.

Следующей разработкой стала БЭСМ (большая электронно-счетная машина), создана в 1952 г., в своем устройстве имела 5 000 электронных ламп, производительность – 10 000 операций в секунду.

В качестве языка программирования в ЭВМ первого поколения использовался машинный язык. Машинный язык – язык программирования, содержание и правила которого реализованы аппаратными средствами ЭВМ. Машинный язык состоит из системы команд ЭВМ и метода кодирования информации (исходных данных, результатов вычислений), принятого в ЭВМ. Символами машинного языка являются двоичные цифры; как правило, символы группируются в конструкции (морфемы) – адреса в командах, коды операций и признаки команд; из команд составляются программы, реализующие алгоритмы задач.

Как правило, ЭВМ первого поколения использовали для научно-технических расчетов, а сам процесс программирования больше напоминал искусство, которым занимался узкий круг математиков, инженеров-электриков и физиков.

### *Второе поколение*

Создание в США 1 июля 1948 г. первого транзистора не предвещало нового этапа в развитии вычислительной техники и ассоциировалось прежде всего с радиотехникой.

На первых порах это был опытный образец нового электронного прибора, требующий серьезного исследования и доработки.

Уже в 1951 г. Уильям Шокли продемонстрировал первый надежный транзистор. Однако стоимость была достаточно велика (до 8 долларов за штуку) и снизилась только после разработки кремниевой технологии, способствовав ускорению процесса миниатюризации в электронике, захватившего и вычислительную технику (рис. 6.2).



*Рисунок 6.2 – Транзисторы*

Еще в 1955 г. была создана бортовая транзисторная ЭВМ для межконтинентальной баллистической ракеты ATLAS и первый полупроводниковый компьютер TRAGIC для ВВС США.

Новая элементная технология позволила резко повысить надежность вычислительной техники, снизить ее габариты и потребляемую мощность, а также значительно повысить производительность. Это позволило создавать ЭВМ с большими логическими возможностями и производительностью, что способствовало распространению сферы применения ЭВМ на решение задач планово-экономических, управления производственными процессами и др. В рамках второго поколения все более четко проявляется дифференциация ЭВМ на малые, средние и большие.

Первой ЭВМ, в которой частично использовали полупроводниковые приборы вместо электронных ламп, была машина SEAC (Standarts Eastern Automatic Computer), созданная в 1951 г. ЭВМ второго поколения – IBM 604, IBM 608, IBM 702 (1953–1955 гг.).

В начале 60-х гг. полупроводниковые машины стали производить и в СССР.

Небольшие отечественные машины второго поколения («Наири», «Раздан», «Мир» и др.) с производительностью порядка  $10^4$  оп/с в конце 60-х годов были доступны каждому вузу.

Отечественные ЭВМ второго поколения серии БЭСМ-6 (1965–1966 гг.) работали до 90-х гг. и по своей производительности – 1 млн операций в секунду приближались к вычислительным машинам третьего поколения.

В качестве языка программирования в ЭВМ второго поколения использовался язык ассемблера – язык программирования низкого уровня, мнемонические команды которого соответствуют инструкциям процессора вычислительной системы. Трансляция программы в исполняемый машинный код производится ассемблером (от англ. *assembler* – сборщик) – программой-транслятором, которая и дала языку ассемблера его название.

Конец 50-х гг. годов характеризуется началом этапа автоматизации программирования, приведшим к появлению языков программирования Fortran, Algol-60 и др. Программа, написанная на известном языке, переводилась на язык команд автоматически, с помощью программы-переводчика – транслятора.

### *Третье поколение*

Третье поколение связывается с появлением ЭВМ с элементной базой на интегральных схемах (рис. 6.3). В январе 1959 г. Джеком Килби была создана первая интегральная схема, представляющая собой тонкую германиевую пластинку длиной в 1 см. Для демонстрации возможностей интегральной технологии фирма Texas Instruments создала для ВВС США бортовой компьютер, содержащий 587 интегральных схем, и объемом ( $40 \text{ см}^3$ ) в 150 раз меньшим, чем у аналогичной ЭВМ старого образца. Но у интегральных схем Килби был ряд существенных недостатков, которые были устранены с появлением в том же году планарных интегральных схем Роберта Нойса.

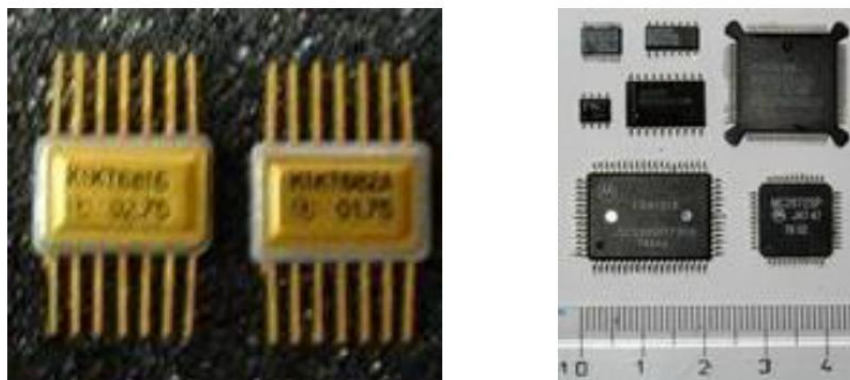


Рисунок 6.3 – Микросхемы

Так, первыми ЭВМ этого поколения стали модели систем IBM (ряд моделей IBM 360) и PDP (PDP 1).

В Советском Союзе в содружестве со странами Совета Экономической Взаимопомощи (Польша, Венгрия, Болгария, ГДР и др.) стали выпускаться модели единой системы (ЕС) и системы малых (СМ) ЭВМ.

Значительно более мощным становится программное обеспечение, обеспечивающее функционирование ЭВМ в различных режимах эксплуатации. Появляются развитые системы управления базами данных (СУБД), системы автоматизирования проектных работ (САПРы); большое внимание уделяется созданию пакетов прикладных программ (ППП) различного назначения. По-прежнему появляются новые и развиваются существующие языки и системы программирования.

#### *Четвертое поколение*

Конструктивно-технологической основой вычислительной техники 4-го поколения становятся большие (БИС) и сверхбольшие (СБИС) интегральные схемы, созданные соответственно в 70–80-х гг. Такие интегральные схемы содержат уже тысячи, десятки и сотни тысяч транзисторов на одном кристалле (чипе). При этом БИС-технология частично использовалась и в проектах предыдущего поколения (IBM/360, ЕС ЭВМ и др.).

Наиболее важный в концептуальном плане критерий, по которому ЭВМ 4-го поколения можно отделить от ЭВМ 3-го поколения, состоит в том, что первые проектировались уже в расчете на эффективное использование современных языков высокого уровня и упрощения процесса программирования для проблемного программиста. В аппаратном отношении для них характерно широкое использование ИС-технологии и быстродействующих запоминающих устройств.

Развитие ЭВМ 4-го поколения пошло по двум направлениям:

1. Создание суперЭВМ – комплексов многопроцессорных машин. Быстродействие таких машин достигает нескольких миллиардов операций в секунду. Они способны обрабатывать огромные массивы информации. Сюда входят комплексы ILLIAS-4, CRAY, CYBER, «Эльбрус-1», «Эльбрус-2» и др.

Многопроцессорные вычислительные комплексы (МВК) «Эльбрус-2» активно использовали в Советском Союзе в областях, требующих большого объема вычислений, прежде всего в оборонной отрасли. Вычислительные комплексы «Эльбрус-2» эксплуатировались в Центре управления космическими полетами, в ядерных исследовательских центрах. Наконец, именно комплексы «Эльбрус-2» с 1991 года использовали в системе противоракетной обороны и на других военных объектах.

2. Дальнейшее развитие на базе БИС и СБИС микро-ЭВМ и персональных ЭВМ (ПЭВМ). Первыми представителями этих машин являются Apple, IBM-PC (XT, AT, PS/2), «Искра», «Электроника», «Мазовия», «Агат», «ЕС-1840», «ЕС-1841» и др.

Начиная с этого поколения ЭВМ повсеместно стали называть компьютерами. А слово «компьютеризация» прочно вошло в наш быт.

Наиболее известной серией ЭВМ четвертого поколения можно считать IBM/370 (рис. 6.4), которая в отличие от не менее известной серии IBM/360 3-го поколения располагает более развитой системой команд и более широким использованием микропрограммирования.

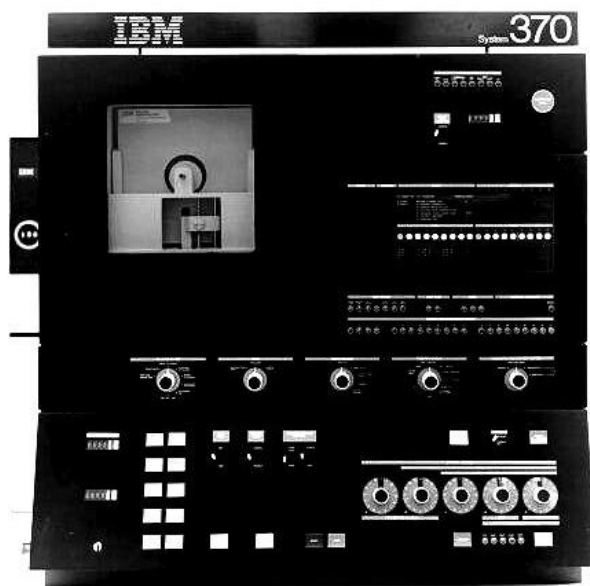
В старших моделях 370-й серии был реализован аппарат виртуальной памяти, позволяющий создавать для пользователя видимость неограниченных ресурсов оперативной памяти.

Феномен персонального компьютера (ПК) восходит к созданию в 1965 г. первой мини-ЭВМ PDP-8. Машина быстро завоевала популярность и стала первым массовым компьютером этого класса. В начале 70-х годов число машин превысило 100 тысяч. Дальнейшим важным шагом был переход от мини- к микро-ЭВМ; этот новый структурный уровень ВТ начал формироваться на рубеже 70-х годов, когда появление БИС дало возможность создать универсальный процессор на одном кристалле.

Первый микропроцессор Intel-4004 был создан в 1971 г. и содержал 2250 элементов, а первый универсальный микропроцессор Intel-8080, явившийся стандартом микрокомпьютерной технологии и созданный в 1974 г., содержал уже 4500 элементов и послужил осно-



вой для создания первых ПК. В 1979 г. выпускается один из самых мощных и универсальных 16-битный микропроцессор Motorola-68000 с 70 000 элементами, а в 1981 г. – первый 32-битный микропроцессор Hewlett Packard с 450 тыс. элементами.



*Рисунок 6.4 – ЭВМ четвертого поколения IBM/370*

Первым ПК можно считать Altair-8800 (рис. 6.5), созданный на базе микропроцессора Intel-8080 в 1974 г. Эдвардом Робертсом.



*Рисунок 6.5 – Персональный компьютер Altair-8800*

Для Altair-8800 Пол Аллен и Бил Гейтс создали транслятор с популярного языка Basic, существенно увеличив интеллектуальность первого ПК (впоследствии они основали знаменитую теперь компанию Microsoft Inc). Через год после появления первого ПК Altair-8800 их в производство ПК включилось более 20 различных компаний и фирм; начала формироваться ПК-индустрия (собственно производст-

во ПК, их сбыт, периодические и непериодические издания, выставки, конференции и т.д.).

В 1981г. фирма IBM, во избежание потери массового рынка, начинает выпуск своих ныне широко известных серий ПК IBM PC/XT/AT (рис. 6.6) и PS/2, открывших новую эпоху персональной вычислительной техники.



*PC XT*



*PC AT*

*Рисунок 6.6 – ПК IBM PC/XT/AT*

Выход на арену ПК-индустрии гиганта IBM ставит производство ПК на промышленную основу, что позволяет решить целый ряд важных для пользователя вопросов (стандартизация, унификация, развитое программное обеспечение и др.).

В качестве языков программирования в ЭВМ четвертого поколения используются объектно-ориентированные языки (Delphi, Си++ и др.)

Первой ЭВМ, открывающей собственно класс супер-ЭВМ, можно считать модель Amdahl 470V16, созданную в 1975 г. и совместимую с IBM-серией. В настоящее время к классу супер-ЭВМ относят модели, имеющие среднее быстродействие не менее 20 мегафлопсов (1 мегафлопс = 1 млн операций в с плавающей точкой в секунду). Первой моделью с такой производительностью явилась во многом уникальная ЭВМ ILLIAC-IV, созданная в 1975 г. в США и имеющая максимальное быстродействие порядка 50 мегафлопсов.

В общеупотребительный лексикон термин «суперкомпьютер» вошел благодаря распространенности компьютерных систем Сеймура Крэя, таких как, CDC 6600, CDC 7600, Cray-1, Cray-2, Cray-3 и Cray-4. Сеймур Крэй разрабатывал вычислительные машины, которые становились основными вычислительными средствами правительственных, промышленных и академических научно-технических проектов США с середины 60-х гг. до 1996 г. Не случайно в то время одним из попу-

лярных определений суперкомпьютера было следующее: «Любой компьютер, который создал Сеймур Крэй». Сам Крэй никогда не называл свои детища суперкомпьютерами, предпочитая использовать вместо этого обычное название «компьютер».

В настоящее время фирма Cray Research является мировым лидером производства современных супер-ЭВМ и сопутствующих средств.

Начиная с 1993 г., суперкомпьютеры ранжируют в списке Top500<sup>8</sup>. Список составляется на основе теста LINPACK по решению системы линейных алгебраических уравнений, являющейся общей задачей для численного моделирования (табл. 6.4).

Таблица 6.4 – Распределение суперкомпьютеров из списка Top500 по странам мира (ноябрь 2021 г.)

Страна	Количество суперкомпьютеров
Китай	173
США	149
Япония	32
Германия	26
Франция	19
Голландия	11
Канада	11
Великобритания	11
Южная Корея	7
Россия	7
Италия	6
Саудовская Аравия	6
Бразилия	5
Швеция	4
Польша	4
Австралия	3

<sup>8</sup>TOP500 List – November 2021 | TOP500. [www.top500.org](http://www.top500.org). (дата обращения 18 ноября 2021)[36].

Самым мощным суперкомпьютером в ноябре 2021 г. по этому списку стал Фугаку, работающий в Центре вычислительных наук Института физико-химических исследований (RIKEN) в Кобе, Япония. Скорость вычислений, производимых им, составляет 442.01 петафлопс (10 в 15 степени вычислительных операций с плавающей запятой в секунду). По этому показателю он в три раза быстрее и в три раза эффективнее предыдущего рекордсмена – Summit, работающего в Ок-Риджской национальной лаборатории в Ок-Ридж, США.

В авторитетном рейтинге суперкомпьютеров Top500 Россия заняла 19 место с новым компьютером «Червоненкис» компании «Яндекс». Впервые за долгое время Россия оказалась в первой двадцатке этого рейтинга. В общей сложности в нем представлено семь российских суперкомпьютера – три «Яндекса», два Сбербанка и по одному МТС и МГУ.

По числу компьютеров в Top500 Россия находится на десятом месте<sup>9</sup>. [26]

#### *Пятое поколение ЭВМ: 1990 – ...*

В соответствии с идеологией развития компьютерных технологий после четвертого поколения, построенного на сверхбольших интегральных схемах, ожидалось создание следующего поколения, ориентированного на распределенные вычисления, одновременно считалось, что пятое поколение станет базой для создания устройств, способных к имитации мышления.

Переход к компьютерам пятого поколения предполагал переход к новым архитектурам, ориентированным на создание искусственного интеллекта.

Считалось, что архитектура компьютеров пятого поколения будет содержать два основных блока. Один из них – собственно компьютер, в котором связь с пользователем осуществляет блок, называемый интеллектуальным интерфейсом. Задача интерфейса – понять текст, написанный на естественном языке или речь, и изложенное таким образом условие задачи перевести в работающую программу.

Широкомасштабная правительственная программа в Японии по развитию компьютерной индустрии и искусственного интеллекта была предпринята в 1980-е гг. Целью программы было создание «эпохального компьютера» с производительностью суперкомпьютера и мощными функциями искусственного интеллекта. Начало разработок – 1982 г., конец разработок – 1992 г., стоимость разработок – 57 млрд йен (по-

---

<sup>9</sup> Россия внезапно ворвалась в мировой топ самых мощных суперкомпьютеров. *CNews.ru*. [https://www.cnews.ru/news/top/2021-11-16\\_rossijskie\\_superkompyutery](https://www.cnews.ru/news/top/2021-11-16_rossijskie_superkompyutery) (дата обращения: 28 мая 2022)[31].

рядка 500 млн долларов). Программа закончилась провалом, так как не опиралась на четкие научные методики, более того, даже ее промежуточные цели оказались недостижимы в технологическом плане.

Основные требования к компьютерам 5-го поколения: создание развитого человеко-машинного интерфейса (распознавание речи, образов); развитие логического программирования для создания баз знаний и систем искусственного интеллекта; создание новых технологий в производстве вычислительной техники; создание новых архитектур компьютеров и вычислительных комплексов.

В настоящий момент термин «пятое поколение» является неопределенным и применяется во многих смыслах, например, при описании систем облачных вычислений.

Завершая экскурс в историю современной вычислительной техники с той или иной детализацией отдельных ее этапов, следует сделать несколько существенных замечаний. Прежде всего, имеет место все более гладкий переход одного поколения ЭВМ к другому, когда идеи нового поколения в той или иной мере созревают и даже реализуются в предыдущем поколении. Особенно это заметно при переходе на ИС-технологии производства вычислительной техники, когда определяющий акцент поколений все больше смещается с элементной базы на другие показатели: логическая архитектура, программное обеспечение, интерфейс с пользователем, сферы приложений и др. Появляется самая разнообразная вычислительная техника, характеристики которой не укладываются в традиционные классификационные рамки; складывается впечатление, что мы находимся в начале универсализации вычислительной техники, когда все ее классы стремятся к нивелированию своих вычислительных возможностей. Многие элементы пятого поколения в той или иной мере характерны и в наши дни.

### **6.3. Типы и назначение компьютеров**

Существование различных типов компьютеров определяется различием задач, для решения которых они предназначены. Приведенное ниже деление очень условно.

Различают:

- суперкомпьютеры;
- специализированные компьютеры-серверы;
- встроенные компьютеры-микропроцессоры;
- персональные компьютеры (ПК).

Для выполнения изначального назначения компьютеров – вычислений – на рубеже 60–70 гг. были созданы специализированные ЭВМ, так называемые суперкомпьютеры.

Суперкомпьютеры – специальный тип компьютеров, создающихся для решения предельно сложных вычислительных задач (составления прогнозов, моделирования сложных явлений, обработки сверхбольших объемов информации).

Компьютер, работающий в локальной или глобальной сети, может специализироваться на оказании информационных услуг другим компьютерам, на обслуживании других компьютеров. Такой компьютер называется сервером (от английского слова *serve* – обслуживать, управлять). Например, в локальной сети один из компьютеров, имеющий скоростной жесткий диск большой емкости, может выполнять функции *файлового сервера*, а другой (к которому присоединен принтер) – *сервера печати*.

Кроме привычных компьютеров с клавиатурами, мониторами, дисководами, сегодняшней мир вещей наполнен компьютерами-невидимками. Микропроцессор представляет компьютер в миниатюре. Массовое распространение микропроцессоры получили и в производстве, там, где управление может быть сведено к отдаче ограниченной последовательности команд. Микропроцессоры незаменимы в современной технике. Например, управление современным двигателем – обеспечение экономии расхода топлива, ограничение максимальной скорости движения, контроль исправности и т. д. – немыслимо без использования микропроцессоров. Еще одной перспективной сферой их использования является бытовая техника.

12 августа 1981 г. корпорация IBM представила первую в мире модель персонального компьютера (ПК, РС). Персональные компьютеры совершили компьютерную революцию в профессиональной деятельности миллионов людей и оказали огромное влияние на все стороны жизни человеческого общества.

### **Вопросы для повторения и самоконтроля**

1. Перечислите этапы развития вычислительной техники.
2. Укажите признаки, отличающие одно поколение вычислительной техники от другого.
3. Первое поколение ЭВМ: элементная база, быстродействие, программное обеспечение.
4. Второе поколение ЭВМ: элементная база, быстродействие, программное обеспечение.

5. Третье поколение ЭВМ: элементная база, быстродействие, программное обеспечение.

6. Четвертое поколение ЭВМ: элементная база, быстродействие, программное обеспечение.

7. Пятое поколение ЭВМ: элементная база, быстродействие, программное обеспечение.

8. Какие существуют типы компьютеров?

### **Задания для самостоятельной работы**

1. Первая ламповая ЭВМ называлась:

- а) Марк-1;
- б) ЭНИАК;
- в) Цузе 1.

2. С историей создания вычислительных машин не связан:

- а) Джордж Буль;
- б) Чарльз Беббидж;
- в) Исаак Ньютон;
- г) Леонардо да Винчи.

3. Первые ЭВМ были созданы в XX веке:

- а) в 40-е гг.;
- б) в 60-е;
- в) в 70-е;
- г) в 80-е.

4. Основной элементной базой ЭВМ четвертого поколения являются:

- а) полупроводники;
- б) электромеханические схемы;
- в) электровакуумные лампы;
- г) СБИС.

### *Ответы на задания для самостоятельной работы*

- 1. б.
- 2. в.
- 3. а.
- 4. в.

## Глава 7. СИСТЕМЫ СЧИСЛЕНИЯ. АРИФМЕТИКА В РАЗЛИЧНЫХ СИСТЕМАХ СЧИСЛЕНИЯ

### 7.1. Системы счисления

Изучение систем счисления, которые используются в компьютерах, важно для понимания того, каким образом производится обработка числовых данных в ЭВМ.

Система счисления – способ записи чисел с помощью заданного набора специальных символов (цифр) и сопоставления этим записям реальных значений.

Все системы счисления можно разделить на непозиционные и позиционные.

В непозиционных системах счисления, которые появились значительно раньше позиционных, смысл каждого символа не зависит от того места, на котором он стоит. Примером такой системы счисления является Римская, в которой для записи чисел используются буквы латинского алфавита. При этом буква I всегда означает единицу, буква – V пять, X – десять, L – пятьдесят, C – сто, D – пятьсот, M – тысячу и т. д. Например, число 264 записывается в виде CCLXIV.

Недостатком непозиционных систем является отсутствие формальных правил записи чисел и, соответственно, арифметических действий с многозначными числами.

В позиционных системах счисления вес каждой цифры изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число.

Правила выполнения вычислений с многозначными числами в позиционной системе счисления были разработаны средневековым математиком Мухамедом аль-Хорезми и в Европе и названы алгоритмами.

В вычислительной технике применяют позиционные системы счисления. Позиционных систем счисления существует множество, и отличаются они друг от друга алфавитом – множеством используемых цифр. Размер алфавита (число цифр в нем) называется основанием системы счисления. Последовательная запись символов алфавита (цифр) изображает число. Позиция символа в изображении числа называется разрядом. Разряду с номером 0 соответствует младший разряд целой части числа. Каждому символу соответствует определенное число, которое меньше основания системы счисления. В зависи-



мости от позиции (разряда) числа значение символа умножается на степень основания, показатель которой равен номеру разряда.

Таким образом, целое положительное число  $A$  в позиционной системе счисления можно представить выражением:

$$A = a_n a_{n-1} \dots a_1 a_0 = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p^1 + a_0 p^0 \quad (7.1)$$

или

$$A = \sum_{k=0}^n a_k p^k,$$

где  $p$  – основание системы счисления, целое положительное число;

$a$  – символ (цифра);

$n$  – номер старшего разряда числа.

Обозначения цифр берутся из алфавита, который содержит  $p$  символов. Каждой цифре соответствует определенный количественный эквивалент. Обозначение  $a_k$  следует понимать как цифру в  $k$ -м разряде. Всегда выполняется неравенство:  $a_k < p$ .

Запись  $A(p)$  указывает, что число  $A$  представлено в системе счисления с основанием  $p$ :

$$A_{(p)} = a_n a_{n-1} \dots a_1 a_0. \quad (7.2)$$

Примером системы счисления является всем нам хорошо известная десятичная система счисления. Любое число в ней записывается с помощью цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Важно, что значение каждой цифры зависит от того места, на котором она стоит в этой записи.

Например, 1575: цифра 5 в записи числа встречается дважды: цифра 5 в последнем разряде – число единиц, а цифра 5, находящаяся в записи числа левее, – число сотен.

В десятичной системе счисления значение единицы каждого разряда в 10 раз больше единицы соседнего с ним правого разряда.

Само число 10 называется основанием системы счисления, а цифры, используемые в десятичной системе – базисными числами этой системы.

Но в качестве основания системы счисления можно выбрать любое целое число.

В компьютере для представления информации используют десятичную, двоичную, восьмеричную и шестнадцатеричную системы счисления. Количество цифр, которое требуется для изображения числа в позиционной системе счисления, равно основанию системы счисления  $p$ .

Например, для записи чисел в двоичной системе счисления требуется две цифры, в десятичной – десять, а в шестнадцатеричной – шестнадцать.

Двоичная система счисления имеет набор цифр  $\{0, 1\}$ ,  $p = 2$ .

В общем виде, используя формулу (7.1), двоичное число можно представить выражением:

$$A_{(2)} = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0. \quad (7.3)$$

Например, число  $101101_{(2)}$  по формуле (7.3) можно записать так:

$$101101_{(2)} = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0.$$

Открытие двоичного способа представления чисел приписывают китайскому императору Фо Ги, жизнь которого относится к IV тысячелетию до новой эры. Известный немецкий математик Лейбниц в 1697 г. разработал правила двоичной арифметики. Он подчеркивал, что «вычисление с помощью двоек, т. е. 0 и 1, в вознаграждение его длиннот, является для науки основным и порождает новые открытия, которые оказываются полезными впоследствии, даже в практике чисел, а особенно в геометрии: причиной чего служит то обстоятельство, что при сведении чисел к простейшим началам, каковы 0 и 1, всюду выявляется чудесный порядок».

Блестящие предсказания Лейбница сбылись только через 2,5 столетия, когда именно двоичная система счисления нашла применение в качестве универсального способа кодирования информации в компьютерах.

Шестнадцатеричная система счисления имеет набор цифр  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ ,  $p = 16$ .

Для изображения чисел в шестнадцатеричной системе счисления требуются 16 цифр. Для обозначения первых десяти цифр используются цифры десятичной системы счисления, шесть остальных – первых шесть прописных букв латинского алфавита. По формуле (7.1) шестнадцатеричное число может быть представлено так:

$$A_{(16)} = a_n \cdot 16^n + a_{n-1} \cdot 16^{n-1} + \dots + a_1 \cdot 16^1 + a_0 \cdot 16^0. \quad (7.4)$$

*Пример.* Число E7F8140 по формуле (7.4) запишется так:

$$E \cdot 16^6 + 7 \cdot 16^5 + F \cdot 16^4 + 8 \cdot 16^3 + 1 \cdot 16^2 + 4 \cdot 16^1 + 0 \cdot 16^0.$$

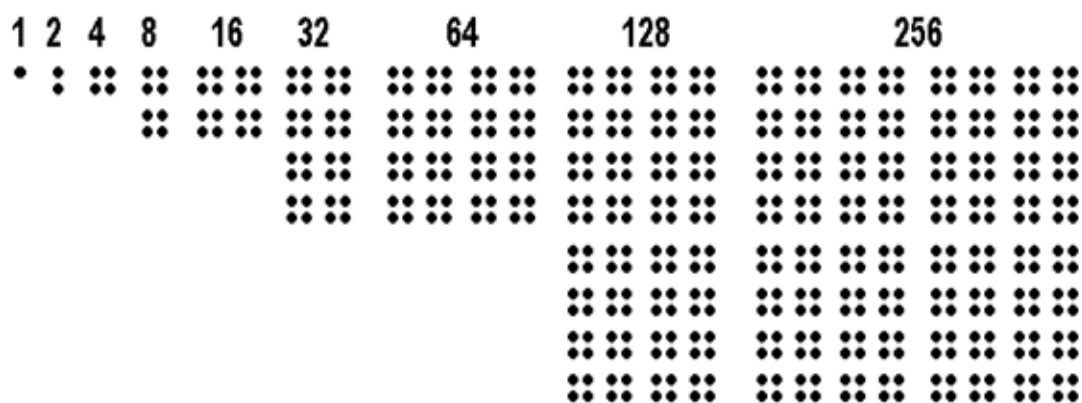
Восьмеричная система счисления имеет набор цифр {0, 1, 2, 3, 4, 5, 6, 7},  $p = 8$ .

Для изображения чисел в восьмеричной системе счисления требуются 8 цифр. Восьмеричная система используется реже, но ее тоже полезно знать.

Представление информации, хранящейся в памяти компьютера, в ее истинном двоичном виде весьма громоздко из-за большого количества цифр.

*Каждый разряд двоичного числа имеет информационную емкость 1 бит.* На основании одного двоичного разряда можно закодировать только два десятичных числа. На основании двух двоичных разрядов можно закодировать уже четыре десятичных числа. Три двоичными разрядами можно представить восемь десятичных чисел и т. д.

Мы видим, что добавление каждого следующего разряда вдвое увеличивает количество двоичных комбинаций. Графически это может быть представлено так (рис. 7.1):



*Рисунок 7.1 – Каждый следующий разряд двоичного числа удваивает количество возможных комбинаций из нулей и единиц*

Поэтому при записи двоичной информации на бумаге или выводе ее на экран принято использовать восьмеричную или шестнадца-

теричную системы счисления. В современных компьютерах чаще используется шестнадцатеричная система счисления.

Полезно помнить некоторые степени двойки, восьмерки и шестнадцати (табл. 7.1).

Таблица 7.1 – Степени двойки, восьмерки и шестнадцати

k	$2^k$	$8^k$	$16^k$
1	2	8	16
2	4	64	256
3	8	512	4096
4	16	4096	65536
5	32	3276	1048576
6	64		
7	128		
8	256		
9	512		
10	1024		
11	2048		
12	4096		

Запись в этих системах счисления первых двух десятков целых чисел приведена в таблице 7.2.

Таблица 7.2 – Соответствие чисел в различных системах счисления

10-я	2-я	8-я	16-я	10-я	2-я	8-я	16-я
0	0	0	0	10	1010	12	A
1	1	1	1	11	1011	13	B
2	10	2	2	12	1100	14	C
3	11	3	3	13	1101	15	D
4	100	4	4	14	1110	16	E
5	101	5	5	15	1111	17	F
6	110	6	6	16	10000	20	10
7	111	7	7	17	10001	21	11
8	1000	10	8	18	10010	22	12
9	1001	11	9	19	10011	23	13

Из всех систем счисления особенно проста и поэтому интересна для технической реализации в компьютерах двоичная.

Компьютеры используют двоичную систему, потому что она имеет ряд преимуществ перед другими системами:

- для ее реализации нужны технические устройства с двумя устойчивыми состояниями (есть ток – нет тока, намагничен – не намагничен и т. п.), а не с десятью, как в десятичной;
- представление информации посредством только двух состояний надежно и помехоустойчиво;
- возможно применение аппарата булевой алгебры для выполнения логических преобразований информации;
- двоичная арифметика намного проще десятичной.

Недостаток двоичной системы – быстрый рост числа разрядов, необходимых для записи чисел (см. табл. 7.2).

## 7.2. Перевод чисел из одной системы счисления в другую

Перевод чисел из одной системы счисления в другую составляет важную часть машинной арифметики.

Рассмотрим основные правила перевода<sup>10</sup>.

### *Правило 1*

Для перевода чисел из любой системы счисления в десятичную систему счисления, можно воспользоваться выражением (7.1). Сначала в десятичную систему счисления переводится основание той системы, из которой осуществляется перевод, а затем цифры исходного числа. Результаты подставляются в выражение (7.1). Полученная сумма дает искомый результат.

### *Пример*

Разряды	3 2 1 0 -1
Число	$1011,1_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 11,5_{10}$ .

Разряды	2 1 0 -1
Число	$276,5_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} = 190,625_{10}$ .

Разряды	2 1 0
Число	$1F3_{16} = 1 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 = 499_{10}$ .

### *Схема Горнера*

Эквивалентными являются алгоритмы для вычисления значения многочлена в некоторой точке  $x$ , заданные следующими формулами:

---

<sup>10</sup>Большинство примеров взято из <http://www.kbsu.ru/~book> [29].

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0; \quad (7.5)$$

$$P(x) = (\dots (((a_n x + a_{n-1})x + a_{n-2})x + a_{n-3})x + \dots + a_1)x + a_0. \quad (7.6)$$

Запись (7.6) носит название вычислительной схемы Горнера.

Например:

$$P(x) = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = (((a_4 x + a_3) x + a_2) x + a_1) x + a_0$$

Алгоритм, задаваемый формулой (7.6), требует меньше арифметических операций и сводится к выполнению последовательной цепочки операций умножения и сложения в порядке их записи слева направо, поэтому при переводе чисел в десятичную систему счисления можно воспользоваться схемой Горнера.

### Правило 2

Для перевода целых чисел из десятичной системы счисления в систему счисления с основанием  $p$  надо последовательно делить заданное число и получаемые целые части на новое основание счисления  $p$  до тех пор, пока целая часть не станет меньше нового основания счисления.

Полученные остатки от деления, представленные цифрами из нового счисления, записать в виде числа, начиная с последней целой части.



Рисунок 7.2 – Перевод числа из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную

Пример: Переведем число 75 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную системы (рис. 7.2):

$$\text{Ответ: } 75_{10} = 1\ 001\ 011_2 = 113_8 = 4B_{16}.$$

### Правило 3

Для перевода правильной десятичной дроби  $F$  в систему счисления с основанием  $p$  необходимо  $F$  умножить на  $p$ , записанное в той же десятичной системе, затем дробную часть полученного произведения снова умножить на  $p$  и т. д., до тех пор, пока дробная часть очередного произведения не станет равной нулю, либо не будет достигнута требуемая точность изображения числа  $F$  в  $p$ -ичной системе. Представлением дробной части числа  $F$  в новой системе счисления будет последовательность целых частей полученных произведений, записанных в порядке их получения и изображенных одной  $p$ -ичной цифрой. Если требуемая точность перевода числа  $F$  составляет  $k$  знаков после запятой, то предельная абсолютная погрешность при этом равняется  $q - (k + 1) / 2$ .

*Пример:* Переведем число 0,36 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную (рис. 7.3):

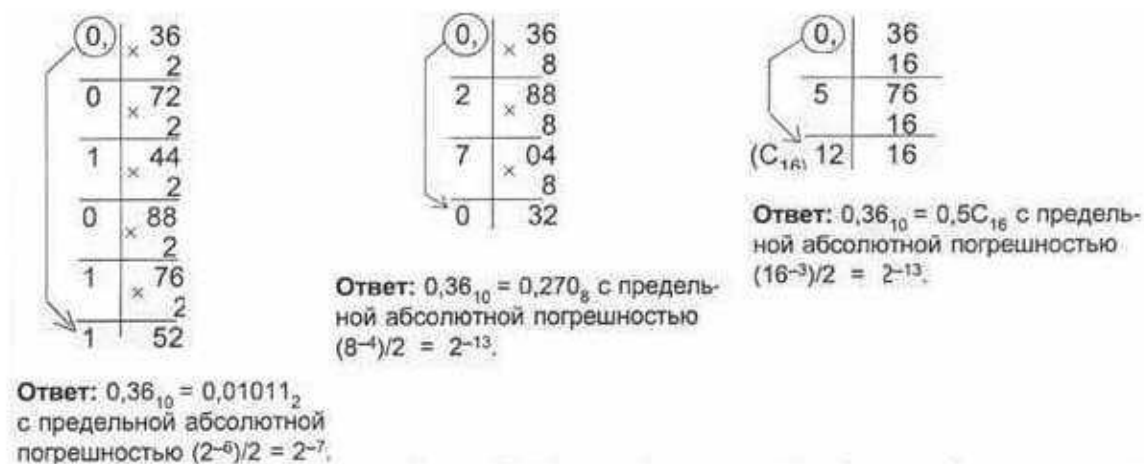


Рисунок 7.3 – Перевод дробного числа из десятичной системы в двоичную, восьмеричную и шестнадцатеричную

Для чисел, имеющих как целую, так и дробную части, перевод из десятичной системы счисления в другую осуществляется отдельно для целой и дробной частей по правилам 2 и 3.

### Правило 4

Преобразования чисел из двоичной в восьмеричную и шестнадцатеричную системы и наоборот просты потому, что числа 8 и 16 являются целыми степенями числа 2.

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему очень прост: достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр). Лишние нули в старших разрядах отбрасываются.

*Пример*

$$537, 1_8 = 101\ 011\ 111, 001_2 ; 1A3, F_{16} = 1\ 1010\ 0011, 1111_2$$

$$\begin{array}{cccccccccccc} \downarrow & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & \downarrow & & & & \\ 5 & 3 & 7 & 1 & & 1 & A & 3 & F & & & & \end{array}$$

*Правило 5*

Обратный перевод производится так: каждая триада двоичных цифр заменяется восьмеричной цифрой, а каждая тетрада двоичных цифр заменяется шестнадцатеричной цифрой. Для правильного перевода число должно быть выровнено, т. е. число двоичных знаков должно быть кратно трем (для восьмеричной цифры) или четырем (шестнадцатеричной цифры). Выравнивание производится простым дописыванием требуемого количества нулей перед старшим разрядом целой части числа.

*Пример*

$$10101001, 10111_2 = 10\ 101\ 001, 101\ 110_2 = 251,56_8$$

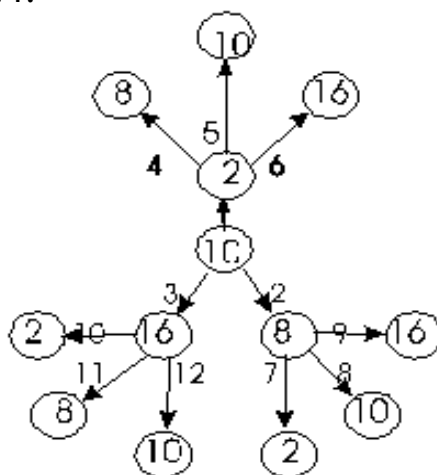
$$\begin{array}{cccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\ 2 & 5 & 1 & 5 & 6 & \end{array}$$

$$10101001, 10111_2 = 1010\ 1001, 1011\ 1000_2 = A9, B8_{16}$$

$$\begin{array}{cccc} \downarrow & \downarrow & \downarrow & \downarrow \\ A & 9 & B & 8 & \end{array}$$

*Сводная таблица переводов целых чисел из одной системы счисления в другую*

Рассмотрим десятичную, двоичную, восьмеричную и шестнадцатеричную системы счисления, которые применяются в компьютерах. Возьмем произвольное десятичное число, например 46, и для него выполним все возможные последовательные переводы из одной системы счисления в другую. Порядок переводов определим в соответствии с рисунком 7.4.



*Рисунок 7.4 – Сводная схема переводов целых чисел из одной системы счисления в другую*



На этом рисунке использованы следующие обозначения:

- в кружках записаны основания систем счисления;
- стрелки указывают направление перевода;
- номер рядом со стрелкой означает порядковый номер соответствующего примера в сводной таблице 7.3.

Таблица 7.3 – Сводная таблица перевода целых чисел

№ п./п	Перевод	№ п./п	Перевод
1	$\begin{array}{r} 10 \rightarrow 2 \\ 46 \overline{) 2} \\ \underline{0} \phantom{0} \\ 23 \phantom{0} \\ \underline{1} \phantom{00} \\ 11 \phantom{00} \\ \underline{1} \phantom{000} \\ 5 \phantom{000} \\ \underline{1} \phantom{0000} \\ 2 \phantom{0000} \\ \underline{0} \phantom{00000} \\ 2 \phantom{00000} \\ \underline{0} \phantom{000000} \\ 1 \phantom{000000} \end{array}$ <p>Ответ: <math>101110_2</math></p>	5	$2 \rightarrow 10$ $101110_2 = 2^5 + 2^3 + 2^2 + 2^1 = 46_{10}$ <p>Ответ: <math>46_{10}</math></p>
		6	$2 \rightarrow 16$ $101110_2 = 10 \underbrace{1110_2}_{2E_{16}} = 2E_{16}$ <p>Ответ: <math>2E_{16}</math></p>
		7	$8 \rightarrow 2$ $56_8 = \underbrace{101}_{5} \underbrace{110_2}_{6}$ <p>Ответ: <math>101110_2</math></p>
2	$\begin{array}{r} 10 \rightarrow 8 \\ 46 \overline{) 8} \\ \underline{6} \phantom{0} \\ 5 \phantom{0} \end{array}$ <p>Ответ: <math>56_8</math></p>	8	$8 \rightarrow 10$ $56_8 = 5 \cdot 8^1 + 6 \cdot 8^0 = 40 + 6 = 46_{10}$ <p>Ответ: <math>46_{10}</math></p>
		9	$8 \rightarrow 16$ $56_8 = \underbrace{101}_{2E_{16}} \underbrace{110_2}_{2E_{16}} = 2E_{16} \underbrace{110_2}_{2E_{16}} = 2E_{16}$ <p>Ответ: <math>2E_{16}</math></p>
3	$\begin{array}{r} 10 \rightarrow 16 \\ 46 \overline{) 16} \\ \underline{14} \phantom{0} \\ 2 \phantom{0} \end{array}$ <p>Ответ: <math>2E_{16}</math></p>	10	$16 \rightarrow 2$ $2E_{16} = \underbrace{0010}_{2} \underbrace{1110_2}_{E} = 101110_2$ <p>Ответ: <math>101110_2</math></p>
		11	$16 \rightarrow 8$ $2E_{16} = 10 \underbrace{1110_2}_{6} = 101 \underbrace{110_2}_{E} = 56_8$ <p>Ответ: <math>56_8</math></p>
4	$2 \rightarrow 8$ $101110_2 = \underbrace{101}_5 \underbrace{110_2}_6 = 56_8$ <p>Ответ: <math>56_8</math></p>	12	$16 \rightarrow 10$ $2E_{16} = 2 \cdot 16^1 + E \cdot 16^0 = 32 + 14 = 46_{10}$ <p>Ответ: <math>46_{10}</math></p>

### 7.3. Арифметические операции, выполняемые в позиционных системах счисления

В вычислительной технике наиболее часто выполняют операции сложения. Вычисления выполняют по следующим *правилам*:

- операция сложения выполняется поразрядно, начиная с младших разрядов в слагаемых;
- в каждом одноименном разряде слагаемых суммируются соответствующие цифры и перенос из предыдущего разряда суммы;
- если сумма цифр одноименных разрядов слагаемых и переноса меньше основания системы счисления, то перенос в следующий разряд равен нулю, если равна или больше, то равен единице.

В качестве примера рассмотрим арифметические операции в десятичной и двоичной системе счисления.

*Арифметические операции над числами в двоичной системе счисления*

Рассмотрим правила выполнения арифметических операций над однозначными числами (табл. 7.4)

Таблица 7.4 – Правила выполнения арифметических операций над однозначными числами (двоичная система счисления)

Правила сложения	Правила вычитания	Правила умножения
$0 + 0 = 0$	$0 - 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$0 - 1 = -1$	$1 \cdot 0 = 0$
$1 + 0 = 1$	$1 - 0 = 1$	$0 \cdot 1 = 0$
$1 + 1 = 10$	$1 - 1 = 0$	$1 \cdot 1 = 1$

#### **Сложение**

*Пример 1.* Сложим числа 15 и 6.

Десятичная:  $15_{10} + 6_{10}$

Двоичная:  $1111_2 + 110_2$

$$\begin{array}{r}
 1 \\
 + 15 \\
 + 6 \\
 \hline
 21 \\
 \hline
 \begin{array}{l}
 5+6=11=10+1 \\
 1+1=2
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 111 \\
 + 1111 \\
 + 0110 \\
 \hline
 10101 \\
 \hline
 \begin{array}{l}
 1+0=1 \\
 1+1=2=2+0 \\
 1+1+1=3=2+1 \\
 1+1=2=2+0
 \end{array}
 \end{array}$$



## Умножение

Выполняя умножение многозначных чисел в различных позиционных системах счисления, можно использовать обычный алгоритм перемножения чисел в столбик, но при этом результаты перемножения и сложения однозначных чисел необходимо заимствовать из соответствующих рассматриваемой системе таблиц умножения и сложения.

Ввиду чрезвычайной простоты таблицы умножения в двоичной системе, умножение сводится лишь к сдвигам множимого и сложениям.

*Пример 5.* Перемножим числа 5 и 6.

Десятичная:  $5_{10} \cdot 6_{10}$       Двоичная:  $101_2 \cdot 110_2$

$$\begin{array}{r} \times 5 \\ 6 \\ \hline 30 \end{array}$$

$$\begin{array}{r} \times 101 \\ 110 \\ \hline 101 \\ 101 \\ \hline 11110 \end{array}$$

*Проверка.* Преобразуем полученные произведения к десятичному виду:  $11110_2 = 2^4 + 2^3 + 2^2 + 2^1 = 30$ .

*Пример 6.* Перемножим числа 115 и 51.

Десятичная:  $115_{10} \cdot 51_{10}$       Двоичная:  $1110011_2 \cdot 110011_2$

$$\begin{array}{r} \times 115 \\ 51 \\ \hline 115 \\ 575 \\ \hline 5865 \end{array}$$

$$\begin{array}{r} \times 1110011 \\ 110011 \\ \hline 1110011 \\ 1110011 \\ \hline 1011011101001 \end{array}$$

*Проверка.* Преобразуем полученные произведения к десятичному виду:  $1011011101001_2 = 2^{12} + 2^{10} + 2^9 + 2^7 + 2^6 + 2^5 + 2^3 + 2^0 = 5865$ .

## 7.4. Кодирование информации

В качестве *наименьшей единицы измерения информации* принят **1 бит**, который соответствует одному разряду в двоичной системе счисления. Эта система лежит в основе архитектуры компьютеров. Для представления всего многообразия величин в компьютере объединяют несколько двоичных разрядов.

Более крупными единицами измерения в компьютере являются:

1 байт = 8 бит

1 Килобайт = 1024 байта =  $2^{10}$  байт

1 Мегабайт = 1024 Кбайта =  $2^{20}$  байт

1 Гигабайт = 1024 Мбайта =  $2^{30}$  байт

1 Терабайт = 1024 Гбайта =  $2^{40}$  байт

1 Петабайт = 1024 Тбайта =  $2^{50}$  байт

1 Экзабайт = 1024 Пбайта =  $2^{60}$  байт

1 Зеттабайт = 1024 Эбайта =  $2^{70}$  байт

1 Йоттабайт = 1024 Збайта =  $2^{80}$  байт

Поскольку информация в компьютере хранится в дискретной форме, для ее записи используется некоторый *конечный набор знаков*, называемый *алфавитом*. Очень часто в качестве алфавита используется таблица кодов, содержащая около 256 знаков. Каждому знаку соответствует числовой код. Этот код хранит образ соответствующего знака в памяти компьютера. Для понимания системы кодирования информации необходимо рассмотреть *правила преобразования числовых кодов в различные системы счисления*.

Наиболее популярна таблица кодов ASCII (рис. 7.5). Она состоит из 16 строк и 16 столбцов, пронумерованных от 0 до F в 16-ричной системе счисления. Например, в столбце 4 и строке D таблицы расположена заглавная буква M латинского алфавита. При записи текста с такой буквой, она будет храниться в памяти в виде кода  $4D_{16}$  или  $77_{10}$ . Обычно последние 8 столбцов таблицы кодов содержат буквы национальных алфавитов, графические знаки. В большом количестве разновидностей таблицы кодов ASCII первая половина таблицы является неизменной, а вторая – переменной.

Таким образом, для хранения одного символа в ASCII-кодировке **требуется 1 байт** памяти компьютера. Однако 8-битовая кодировка является недостаточной для кодировки всех символов расширенных алфавитов. Все препятствия могут быть сняты при переходе на 16-битовую кодировку *Unicode*, допускающую 65536 кодовых комбинаций.

Числа кодируются особым образом.

# Таблица ASCII кодов

Основная таблица ASCII

	00	10	20	30	40	50	60	70
0		▸		0	1	2	3	4
1	␣	␣	␣	␣	␣	␣	␣	␣
2	␣	␣	␣	␣	␣	␣	␣	␣
3	␣	␣	␣	␣	␣	␣	␣	␣
4	␣	␣	␣	␣	␣	␣	␣	␣
5	␣	␣	␣	␣	␣	␣	␣	␣
6	␣	␣	␣	␣	␣	␣	␣	␣
7	␣	␣	␣	␣	␣	␣	␣	␣
8	␣	␣	␣	␣	␣	␣	␣	␣
9	␣	␣	␣	␣	␣	␣	␣	␣
A	␣	␣	␣	␣	␣	␣	␣	␣
B	␣	␣	␣	␣	␣	␣	␣	␣
C	␣	␣	␣	␣	␣	␣	␣	␣
D	␣	␣	␣	␣	␣	␣	␣	␣
E	␣	␣	␣	␣	␣	␣	␣	␣
F	␣	␣	␣	␣	␣	␣	␣	␣

Расширенная таблица ASCII (cp866)

	00	10	20	30	40	50	60	70
0	А	В	С	Д	Е	Ж	З	И
1	Й	К	Л	М	Н	О	П	Р
2	С	Т	У	Ф	Х	Ц	Ч	Ш
3	Щ	Ъ	Ы	Ь	Э	Ю	Я	
4	␣	␣	␣	␣	␣	␣	␣	␣
5	␣	␣	␣	␣	␣	␣	␣	␣
6	␣	␣	␣	␣	␣	␣	␣	␣
7	␣	␣	␣	␣	␣	␣	␣	␣
8	␣	␣	␣	␣	␣	␣	␣	␣
9	␣	␣	␣	␣	␣	␣	␣	␣
A	␣	␣	␣	␣	␣	␣	␣	␣
B	␣	␣	␣	␣	␣	␣	␣	␣
C	␣	␣	␣	␣	␣	␣	␣	␣
D	␣	␣	␣	␣	␣	␣	␣	␣
E	␣	␣	␣	␣	␣	␣	␣	␣
F	␣	␣	␣	␣	␣	␣	␣	␣

Рисунок 7.5 – Таблица кодов ASCII

## 7.4.1. Представление чисел в компьютере

Теперь рассмотрим более подробно процесс кодирования чисел в компьютере. Среди чисел, которые мы используем, встречаются натуральные, целые, рациональные, иррациональные.

В вычислительных машинах применяются две формы представления чисел:

- естественная форма или форма с фиксированной запятой (точкой);
- нормализованная форма или форма с плавающей запятой (точкой).

С фиксированной запятой числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной.

Например, 32,54; 0,0036; –108,2. Эта форма проста, естественна, но имеет небольшой диапазон представления чисел и поэтому не всегда приемлема при вычислениях. Если в результате операции полу-

чится число, выходящее за допустимый диапазон, происходит переполнение разрядной сетки и дальнейшие вычисления теряют смысл. В современных компьютерах форма представления чисел с фиксированной запятой используется только для целых чисел.

С плавающей запятой числа изображаются в виде  $X = \pm M \times P^{\pm r}$ , где  $M$  – мантисса числа (правильная дробь в пределах  $0,1 \leq M < 1$ ),  $r$  – порядок числа (целое),  $P$  – основание системы счисления.

Например, приведенные выше числа с фиксированной запятой можно преобразовать в числа с плавающей запятой так:  $0,3254 \times 10^2$ ,  $0,36 \times 10^{-2}$ ,  $-0,1082 \times 10^3$ . Нормализованная форма представления имеет огромный диапазон чисел и является основной в современных ЭВМ.

Всякое десятичное число, прежде чем оно попадает в память компьютера, преобразуется по схеме:

$$X_{10} \rightarrow X_2 \rightarrow X_2 = M_2 \times 10_2^r$$

После этого осуществляется еще одна важная процедура:

- мантисса с ее знаком заменяется кодом мантиссы с ее знаком;
- порядок числа с его знаком заменяется кодом порядка с его знаком.

Указанные коды двоичных чисел – это образы чисел, которые и воспринимают вычислительные устройства.

**Каждому двоичному числу можно поставить в соответствие несколько видов кодов.**

Существуют следующие коды двоичных чисел.

**Прямой код.** Прямой код двоичного числа (а это либо мантисса, либо порядок) образуется по такому алгоритму:

1. Определить данное двоичное число: оно либо целое (порядок), либо правильная дробь (мантисса).
2. Если это дробь, то цифры после запятой можно рассматривать как целое число.

3. Если это целое и положительное двоичное число, то вместе с добавлением 0 в старший разряд число превращается в код. Для отрицательного двоичного числа перед ним ставится единица. *Например:*

число  $X_2 = -0,101101_2 \rightarrow$  код числа  $X_{пр} = 1101101$ ;

число  $Y_2 = +0,1101101_2 \rightarrow$  код числа  $Y_{пр} = 01101101$ .

Жирным шрифтом выделены знаковые разряды и, кроме того, у кодов отсутствует индекс «2».

**Обратный код.** Обратный код положительного двоичного числа совпадает с прямым кодом, а для отрицательного числа нужно, ис-

ключая знаковый разряд, во всех остальных разрядах нули заменить на единицы и наоборот.

*Например:*

число  $X_2 = -0,10101_2 \rightarrow X_{пр} = 1\ 10101 \rightarrow X_{обр} = 101010$ ;

число  $Y_2 = +0,1101_2 \rightarrow Y_{пр} = 01101 = Y_{обр}$ .

**Дополнительный код.** Дополнительный код положительного числа совпадает с его прямым кодом. Дополнительный код отрицательного числа образуется путем прибавления 1 к обратному коду.

*Например:*

число  $X_2 = -0,10010_2 \rightarrow X_{пр} = 110010 \rightarrow X_{обр} = 101101 \rightarrow X_{доп} = 101110$ ;

число  $Y_2 = +0,1011 \rightarrow Y_{пр} = 01011 = Y_{обр} = Y_{доп}$ .

#### 7.4.2. Сложение и вычитание двоичных чисел

Сложение чисел, а также вычитание чисел в обратном или дополнительном кодах выполняется с использованием обычного правила арифметического сложения многоразрядных чисел. Это правило распространяется и на знаковые разряды чисел.

Различие же обратного и дополнительного кодов связано с тем, что потом делают с единицей переноса из старшего разряда, изображающего знак числа. При сложении чисел в обратном коде эту единицу надо прибавить к младшему разряду результата, а в дополнительном коде единица переноса из старшего разряда игнорируется. Это очевидно, если вспомнить, что дополнительный код из обратного получается как раз прибавлением единицы. Рассмотрим *пример*: сложить числа +18 и -7 (табл. 7.5).

Таблица 7.5 – Таблица сложения чисел в прямом и дополнительном кодах

	В обратном коде		В дополнительном коде	
Десятичная форма	+18	-7	+18	-7
Двоичная форма	+10010	-111	+10010	-111
Прямой код	00010010	10000111	00010010	10000111
Обратный код	00010010	11111000		
Дополнительный код			00010010	11111001
Сложение	00010010		00010010	
	11111000		11111000	
	-----		-----	
Прибавить	(1)00001010 + 1	Игнорировать	(1)00001011	



Итак, как в обратном, так и в дополнительном кодах результаты сложения совпали и равны  $1011_2 = 1110$ .

Умножение и деление двоичных чисел в ЭВМ производится в прямом коде, а их знаки используются лишь для определения знака результата. Также как и в математике, умножение и деление сводится к операциям сдвигов и сложений (с учетом знака числа).

Полученные коды мантииссы и порядка для каждого числа помещаются в ячейки памяти ЭВМ. Для каждой цифры, входящей в код, в ячейке памяти отводится свое отдельное место. Одна ячейка памяти состоит из 8 бит, т. е. 1 байта.

В современных компьютерах 2 байта выделяются для одного машинного слова. В последних моделях ПК обработка информации ведется двойными словами, содержащими 4 байта. Числа с фиксированной запятой имеют формат одного слова, а числа с плавающей запятой – формат двойного слова.

*Пример:* Дано число с плавающей запятой величиной –  $0,625 \times 10^8$ . Нужно преобразовать его в машинный код и заполнить 32-разрядную ячейку памяти. Мантиисса числа  $0,625_{10} = 0,101_2$ .

Поскольку порядок чисел может быть как положительным, так и отрицательным, то машинный порядок смещается относительно естественного таким образом, чтобы весь диапазон машинных порядков изменялся от 0 до максимума, определяемого количеством разрядов, выделяемых для размещения чисел порядка.

Обычно в 32-разрядной ячейке цифры порядка занимают 7 разрядов старшего байта, а восьмой разряд используется для фиксации знака числа. Семь двоичных разрядов позволяют разместить диапазон десятичных разрядов от  $-64$  до  $+63$ .

Если обозначить машинный порядок через  $R$ , а естественный через  $r$ , то связь между ними будет такая:  $R_{10} = r_{10} + 64_{10}$ .

Для двоичной системы счисления  $R_2 = r_2 + 1000000_2$ . В нашем примере порядок  $r$  равен  $8_{10} = 1000_2$ , следовательно,  $R_2 = 1001000$ .

В двоичной системе исходное число выглядит так:  $-0,101 \times 10^{1000}$ . Запись этого числа в 32-разрядной ячейке представлена в таблице 7.6.

Таблица 7.6 – Сводная таблица переводов целых чисел

	Знак числа	П о р я д о к							М а н т и с с а								
Число	1	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	
№ разряда	31	30	29	28	27	26	25	24	23	22	21	20	19	18	...	1	0

### Вопросы для повторения и самоконтроля

1. Для чего нужно изучать системы счисления, которые используются в компьютере?
2. Что называется системой счисления?
3. На какие два типа можно разделить все системы счисления?
4. Какие системы счисления называются непозиционными? Почему? Приведите пример такой системы счисления и записи чисел в ней?
5. Какие системы счисления применяют в вычислительной технике: позиционные или непозиционные? Почему?
6. Какие системы счисления называют позиционными?
7. Как изображается число в позиционной системе счисления?
8. Что называется основанием системы счисления?
9. Что называется разрядом в изображении числа?
10. Как можно представить целое положительное число в позиционной системе счисления?
11. Приведите пример позиционной системы счисления.
12. Опишите правила записи чисел в десятичной системе счисления: а) какие символы образуют алфавит десятичной системы счисления? б) что является основанием десятичной системы счисления? в) как изменяется вес символа в записи числа в зависимости от занимаемой позиции?
13. Какие числа можно использовать в качестве основания системы счисления?
14. Какие системы счисления применяют в компьютере для представления информации?
15. Охарактеризуйте двоичную систему счисления: алфавит, основание системы счисления, запись числа.

16. Почему двоичная система счисления используется в информатике?

17. Дайте характеристику шестнадцатеричной системе счисления: алфавит, основание, запись чисел. Приведите примеры записи чисел.

18. Дайте характеристику восьмеричной системе счисления: алфавит, основание, запись чисел. Приведите примеры записи чисел.

19. По каким правилам выполняется сложение двух положительных целых чисел?

20. Каковы правила выполнения арифметических операций в двоичной системе счисления?

21. Для чего используют перевод чисел из одной системы счисления в другую?

22. Сформулируйте правила перевода чисел из системы счисления с основанием  $p$  в десятичную систему счисления и обратного перевода: из десятичной системы счисления в систему счисления с основанием  $p$ . Приведите примеры.

23. В каком случае для перевода чисел из одной системы счисления в другую может быть использована схема Горнера вычисления значения многочлена в точке? Каковы преимущества ее использования перед другими методами? Приведите пример.

24. Как выполнить перевод чисел из двоичной системы счисления в восьмеричную и обратно? Из двоичной системы счисления в шестнадцатеричную и обратно? Приведите примеры. Почему эти правила так просты?

25. По каким правилам выполняют перевод из восьмеричной в шестнадцатеричную системы счисления и наоборот? Приведите примеры.

26. Напишите таблицы правил выполнения двоичного сложения, вычитания и умножения.

27. Какая форма представления информации называется непрерывной, а какая – дискретной?

28. Какая форма представления информации – непрерывная или дискретная – приемлема для компьютеров и почему?

29. Какова единица измерения информации?

30. Как задаются производные единицы измерения информации?

31. Как определяют алфавит?

32. Как кодируют символы в памяти компьютера?

33. Что собой представляет таблица ASCII кодов?
34. Как кодируют целые положительные числа в памяти компьютера?
35. Каковы особенности представления целых отрицательных чисел в памяти компьютера?
36. Как кодируют действительные числа?

### Задания для самостоятельной работы

1. Используя Правило Счета, запишите первые 20 целых чисел в десятичной, двоичной, троичной, пятеричной и восьмеричной системах счисления.

2. Какие целые числа следуют за числами:

- |                 |               |                  |
|-----------------|---------------|------------------|
| а) $1_2$ ;      | е) $1_8$ ;    | п) $F_{16}$ ;    |
| б) $101_2$ ;    | ж) $7_8$ ;    | м) $1F_{16}$ ;   |
| в) $111_2$ ;    | з) $37_8$ ;   | н) $FF_{16}$ ;   |
| г) $1111_2$ ;   | и) $177_8$ ;  | о) $9AF9_{16}$ ; |
| д) $101011_2$ ; | к) $7777_8$ ; | п) $CDEF_{16}$ ? |

3. Какие целые числа предшествуют числам:

- |                |               |                  |
|----------------|---------------|------------------|
| а) $10_2$ ;    | е) $10_8$ ;   | л) $10_{16}$ ;   |
| б) $1010_2$ ;  | ж) $20_8$ ;   | м) $20_{16}$ ;   |
| в) $1000_2$ ;  | з) $100_8$ ;  | н) $100_{16}$ ;  |
| г) $10000_2$ ; | и) $110_8$ ;  | о) $A10_{16}$ ;  |
| д) $10100_2$ ; | к) $1000_8$ ; | п) $1000_{16}$ ? |

4. Какой цифрой заканчивается четное двоичное число? Какой цифрой заканчивается нечетное двоичное число? Какими цифрами может заканчиваться четное троичное число?

5. Какое наибольшее десятичное число можно записать тремя цифрами:

- а) в двоичной системе;
- б) в восьмеричной системе;
- в) в шестнадцатеричной системе?

6. В какой системе счисления справедливо следующее:

- а)  $20 + 25 = 100$ ;
- б)  $22 + 44 = 110$ ?

7. Десятичное число 59 эквивалентно числу 214 в некоторой другой системе счисления. Найдите основание этой системы.

8. Переведите числа в десятичную систему, а затем проверьте результаты, выполнив обратные переводы:

- |                    |                 |                    |
|--------------------|-----------------|--------------------|
| а) $1011011_2$ ;   | е) $517_8$ ;    | л) $1F_{16}$ ;     |
| б) $10110111_2$ ;  | ж) $1010_8$ ;   | м) $ABC_{16}$ ;    |
| в) $011100001_2$ ; | з) $1234_8$ ;   | н) $1010_{16}$ ;   |
| г) $0,1000110_2$ ; | и) $0,34_8$ ;   | о) $0,A4_{16}$ ;   |
| д) $110100,11_2$ ; | к) $123,41_8$ ; | п) $1DE,C8_{16}$ . |

9. Переведите числа из десятичной системы в двоичную, восьмеричную и шестнадцатеричную, а затем проверьте результаты, выполнив обратные переводы:

- а)  $125_{10}$ ; б)  $229_{10}$ ; в)  $88_{10}$ ; г)  $37,25_{10}$ ; д)  $206,125_{10}$ .

10. Переведите числа из двоичной системы в восьмеричную и шестнадцатеричную, а затем проверьте результаты, выполнив обратные переводы:

- |                             |                           |
|-----------------------------|---------------------------|
| а) $100111110111,0111_2$ ;  | г) $1011110011100,11_2$ ; |
| б) $1110101011,1011101_2$ ; | д) $10111,111110111_2$ ;  |
| в) $10111001,101100111_2$ ; | е) $1100010101,11001_2$ . |

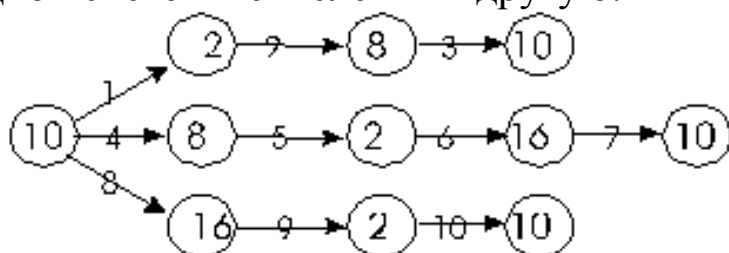
11. Переведите в двоичную и восьмеричную системы шестнадцатеричные числа:

- а)  $2CE_{16}$ ; б)  $9F40_{16}$ ; в)  $ABCDE_{16}$ ; г)  $1010,101_{16}$ ; д)  $1ABC,9D_{16}$ .

12. Выпишите целые числа:

- а) от  $101101_2$  до  $110000_2$  в двоичной системе;  
 б) от  $202_3$  до  $1000_3$  в троичной системе;  
 в) от  $14_8$  до  $20_8$  в восьмеричной системе;  
 г) от  $28_{16}$  до  $30_{16}$  в шестнадцатеричной системе.

13. Для десятичных чисел 47 и 79 выполните цепочку переводов из одной системы счисления в другую:



14. Составьте таблицы сложения однозначных чисел в троичной и пятеричной системах счисления.

15. Составьте таблицы умножения однозначных чисел в троичной и пятеричной системах счисления.

16. Сложите числа, а затем проверьте результаты, выполнив соответствующие десятичные сложения:

- а)  $1011101_2$  и  $1110111_2$ ;      д)  $37_8$  и  $75_8$ ;      и)  $A_{16}$  и  $F_{16}$ ;  
 б)  $1011,101_2$  и  $101,011_2$ ;      е)  $165_8$  и  $37_8$ ;      к)  $19_{16}$  и  $C_{16}$ ;  
 в)  $1011_2$ ,  $11_2$  и  $111,1_2$ ;      ж)  $7,5_8$  и  $14,6_8$ ;      л)  $A, B_{16}$  и  $E, F_{16}$ ;  
 г)  $1011_2$ ,  $11,1_2$  и  $111_2$ ;      з)  $6_8$ ,  $17_8$  и  $7_8$ ;      м)  $E_{16}$ ,  $9_{16}$  и  $F_{16}$ .

17. В каких системах счисления выполнены следующие сложения? Найдите основания каждой системы:

а) $\begin{array}{r} 98 \\ + 89 \\ \hline 121 \end{array}$	б) $\begin{array}{r} 1345 \\ + 2178 \\ \hline 3523 \end{array}$	в) $\begin{array}{r} 10101 \\ + 1111 \\ \hline 1011 \\ \hline 20000 \end{array}$	г) $\begin{array}{r} 765 \\ + 576 \\ \hline 677 \\ \hline 2462 \end{array}$	д) $\begin{array}{r} 9E \\ + 56 \\ \hline 79 \\ \hline 167 \end{array}$
--	---	--	---	---

18. Найдите те подстановки десятичных цифр вместо букв, которые делают правильными выписанные результаты (разные цифры замещаются разными буквами):

а) $\begin{array}{r} ABCD \\ + ABCD \\ \hline BDCEC \end{array}$	б) $\begin{array}{r} A \\ + AB \\ \hline ABC \\ \hline BCB \end{array}$	в) $\begin{array}{r} ABCDA \\ + FLCDA \\ \hline FLC LMN \end{array}$	г) $\begin{array}{r} ABCD \\ + EFBCA \\ \hline GHGCIJ \end{array}$	д) $\begin{array}{r} ABCD \\ + ABCEF \\ \hline EGDHIG \end{array}$
--	---	--	--	--

19. Вычтите:

- а)  $111_2$  из  $10100_2$ ;      д)  $15_8$  из  $20_8$ ;      и)  $1A_{16}$  из  $31_{16}$ ;  
 б)  $10,11_2$  из  $100,1_2$ ;      е)  $47_8$  из  $102_8$ ;      к)  $F9E_{16}$  из  $2A30_{16}$ ;  
 в)  $111,1_2$  из  $10010_2$ ;      ж)  $56,7_8$  из  $101_8$ ;      л)  $D,1_{16}$  из  $B,92_{16}$ ;  
 г)  $10001_2$  из  $1110,11_2$ ;      з)  $16,54_8$  из  $30,01_8$ ;      м)  $ABC_{16}$  из  $5678_{16}$ .

20. Перемножьте числа, а затем проверьте результаты, выполнив соответствующие десятичные умножения:

- а)  $101101_2$  и  $101_2$ ;      д)  $37_8$  и  $4_8$ ;  
 б)  $111101_2$  и  $11,01_2$ ;      е)  $16_8$  и  $7_8$ ;  
 в)  $1011,11_2$  и  $101,1_2$ ;      ж)  $7,5_8$  и  $1,6_8$ ;  
 г)  $101_2$  и  $1111,001_2$ ;      з)  $6,25_8$  и  $7,12_8$ .

21. Разделите  $100101102$  на  $10102$  и проверьте результат, умножая делитель на частное.

22. Разделите  $100110101002$  на  $11002$  и затем выполните соответствующее десятичное и восьмеричное деление.

23. Вычислите значения выражений:

- а)  $256_8 + 10110_2$ ;  
 б)  $(60_8 + 12_{10}) - 1F_{16}$ ;  
 в)  $1AD_{16} - 100101100_2 : 1010_2 + 217_8$ ;  
 г)  $1011_2 \cdot 1100_2 : 14_8 + (100000_2 - 40_8)$ .

24. Расположите следующие числа в порядке возрастания:

- а) 748, 1100102, 7010, 3816;
- б) 6E16, 1428, 11010012, 10010;
- в) 7778, 1011111112, 2FF16, 50010;
- г) 10010, 11000002, 6016, 1418.

25. Прочитайте стихотворение. Переведите встречающиеся в нем числительные из двоичной системы счисления в десятичную.

*Необыкновенная девчонка*

*А. Н. Стариков*

Ей было тысяча сто лет,	Она ловила каждый звук
Она в 101-й класс ходила,	Своими десятью ушами,
В портфеле по сто книг носила –	И десять загорелых рук
Все это правда, а не бред.	Портфель и поводок держали.
Когда, пыля десятком ног,	И десять темно-синих глаз
Она шагала по дороге,	Рассматривали мир привычно,...
За ней всегда бежал щенок	Но станет все совсем обычным,
С одним хвостом, зато стоногий.	Когда поймете наш рассказ.

26. За праздничным столом собрались 4 поколения одной семьи: дед, отец, сын и внук. Их возраст в различных системах счисления записывается так 88 лет, 66 лет, 44 года и 11 лет.

Сколько им лет в десятичной системе счисления, если через год их возраст в тех системах счисления можно будет записать как 100?

*Ответы к заданиям для самостоятельной работы*

1. в) троичная: 0, 1, 2, 10, 11, 12, 20, 21, 22, 100, 101, 102, 110, 111, 112, 120, 121, 122, 200, 201; г) пятеричная: 0, 1, 2, 3, 4, 10, 11, 12, 13, 14, 20, 21, 22, 23, 24, 30, 31, 32, 33, 34.

2. а)  $10_2$ ; б)  $110_2$ ; в)  $1000_2$ ; г)  $10000_2$ ; д)  $101100_2$ ; е)  $2_8$ ; ж)  $10_8$ ; з)  $40_8$ ; и)  $200_8$ ; к)  $10000_8$ ; л)  $10_{16}$ ; м)  $20_{16}$ ; н)  $100_{16}$ ; о)  $9AFA_{16}$ ; п)  $CDF0_{16}$ .

3. а)  $1_2$ ; б)  $1001_2$ ; в)  $111_2$ ; г)  $1111_2$ ; д)  $10011_2$ ; е)  $7_8$ ; ж)  $17_8$ ; з)  $77_8$ ; и)  $107_8$ ; к)  $777_8$ ; л)  $F_{16}$ ; м)  $1F_{16}$ ; н)  $FF_{16}$ ; о)  $A0F_{16}$ ; п)  $FFF_{16}$ .

4. Четное двоичное число оканчивается цифрой 0, нечетное двоичное – цифрой 1, четное троичное – цифрами 0, 1 или 2.

5. а) 7; б) 511; в) 4091.

6. а) ни в какой; б) в шестеричной.

7. Основание 5.





18. в)  $A = 9, B = 4, C = 5, D = 3, F = 1, L = 0, M = 7, N = 8$ ;  
 г)  $A = 3, B = 6, C = 2, D = 5, E = 9, F = 7, G = 1, H = 0, I = 4, J = 8$ ;  
 д)  $A = 9, B = 3, C = 4, D = 2, E = 1, F = 8, G = 0, H = 7, I = 6$ .
19. а)  $1101_2$ ; б)  $1,11_2$ ; в)  $1010,1_2$ ; г)  $-10,01_2$ ; д)  $3_8$ ; е)  $33_8$ ; ж)  $22,1_8$ ;  
 з)  $11,25_8$ ; и)  $17_{16}$ ; к)  $1A92_{16}$ ; л)  $-1,7E_{16}$ ; м)  $4BBC_{16}$ .
20. а)  $11100001_2$ ; б)  $11000110,01_2$ ; в)  $1000000,101_2$ ; г)  $1001011,101_2$ ; д)  $174_8$ ; е)  $142_8$ ; ж)  $15.26_8$ ; з)  $55.2222_8$ .
21.  $1111_2$ .
22.  $1100111_2$ ;  $103_{10}$ ;  $147_8$ .
23. а)  $196_{10}$ ; б)  $103_{10}$ ; в)  $542_{10}$ ; г)  $11_{10}$ .
24. а)  $110010_2, 38_{16}, 74_8, 70_{10}$ ; б)  $142_8, 100_{10}, 1101001_2, 6E_{16}$ ;  
 в)  $10111111_2, 500_{10}, 777_8, 2FF_{16}$ ; г)  $1100000_2, 60_{16}, 141_8, 100_{10}$ .
25.  $1100_2 = 12_{10}$ ;  $101_2 = 5_{10}$ ;  $100_2 = 4_{10}$ ;  $10_2 = 2_{10}$ .
26. Дед:  $88_9 = 80_{10}$ ; отец:  $66_7 = 48_{10}$ ; сын:  $44_5 = 24_{10}$ ; внук:  $11_2 = 3_{10}$ .

## Глава 8. АЛГЕБРА ЛОГИКИ

### 8.1. Возникновение логики как самостоятельной науки

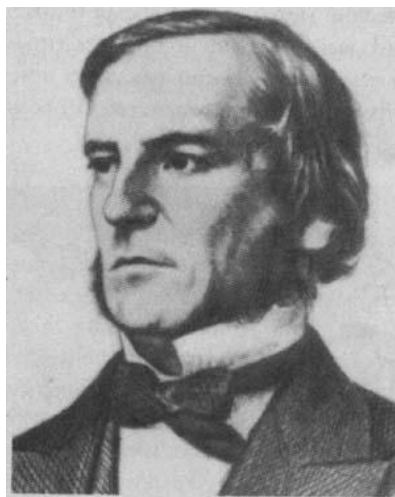
Понятие логики как науки появилось еще в XIX в., т. е. задолго до появления информатики и компьютеров. Элементы математической логики можно найти уже в работах древнегреческих философов.

Логику, основанную Аристотелем (384–322 до н. э.), принято называть формальной. Это название закрепилось за ней, потому что она возникла и развилась как наука о формах мышления.

В XVII в. Г.В. Лейбниц высказал идею о том, что рассуждения могут быть сведены к механическому выполнению определенных действий по установленным правилам. Однако как самостоятельный раздел математики логика начала формироваться только с середины XIX в.

Для того чтобы рассуждать, человеку необходим какой-либо язык. Неудивительно, что математическая логика начиналась с анализа того, как говорят и пишут люди на естественных языках. Выяснилось, что существуют формулировки, которые невозможно разделить на истинные и ложные, но при этом они выглядят осмысленно. Это приводило к возникновению парадоксов, в том числе в одной из фундаментальных наук математики. Тогда было решено создать искусственные формальные языки, лишенные «вольностей» естественного языка.

Джордж Буль (1815–1864 г.) по праву считается отцом математической логики (рис. 8.1). Его именем назван раздел математической логики – булева алгебра.



*Рисунок 8.1 – Джордж Буль – английский математик-самоучка*

Буль изобрел систему обозначений и правил, применимую ко всевозможным объектам, от чисел до предложений. Пользуясь этой системой, он мог закодировать высказывания (утверждения, истинность или ложность которых требовалось доказать) с помощью символов своего языка, а затем манипулировать ими, подобно тому как в математике манипулируют числами.

Через некоторое время стало понятно, что система Буля хорошо подходит для описания электрических переключателей схем. Ток в цепи может либо протекать, либо отсутствовать, подобно тому как утверждение может быть либо истинным, либо ложным.

А еще несколько десятилетий спустя, уже в XX столетии, ученые объединили созданный Джорджем Булем математический аппарат с двоичной системой счисления, заложив тем самым основы для разработки цифрового электронного компьютера.

В 1936 г. выпускник Мичиганского университета Клод Шеннон (1916–2001 г.) сумел ликвидировать разрыв между алгебраической теорией логики и ее практическим применением (рис. 8.2).



*Рисунок 8.2 – Клод Шеннон – американский математик*

Шеннон, имея два диплома бакалавра – по электротехнике и по математике, – выполнял обязанности оператора на неуклюжем механическом вычислительном устройстве под названием дифференциальный анализатор.

Постепенно у Шеннона стали вырисовываться контуры устройства компьютера. Если построить электрические цепи в соответствии с принципами булевой алгебры, то они могли бы выражать логические отношения, определять истинность утверждений, а также выполнять сложные вычисления. Свои идеи относительно связи между двоичным ис-

числением, булевой алгеброй и электрическими схемами Шеннон развил в докторской диссертации, опубликованной в 1938 г.

*Применение в вычислительной технике и информатике алгебры логики*

После изготовления первого компьютера стало ясно, что при его производстве возможно использование только цифровых технологий – ограничение сигналов связи единицей и нулем для большей надежности и простоты архитектуры ПК. Благодаря своей бинарной природе, математическая логика получила широкое распространение в вычислительной технике и информатике. Были созданы электронные эквиваленты логических функций, что позволило применять методы упрощения булевых выражений к упрощению электрической схемы. Кроме того, благодаря возможности нахождения исходной функции по таблице позволило сократить время поиска необходимой логической схемы.

В программировании логика незаменима как строгий язык и служит для описания сложных утверждений, значение которых может определить компьютер.

*Математический аппарат алгебры логики* очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку *основной системой счисления* в компьютере является *двоичная*, в которой используются цифры 1 и 0, а значений логических переменных тоже два: «1 – истина» и «0 – ложь».

Из этого следует два вывода:

1. Одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных.

2. На этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьютера, и, следовательно, уменьшить число элементарных логических элементов компьютера.

Логический элемент компьютера – это часть электронной логической схемы, которая реализует элементарную логическую функцию.

Логическими элементами компьютеров являются электронные схемы И, ИЛИ, НЕ, И–НЕ, ИЛИ–НЕ и другие (называемые также вентилями), а также триггер. Триггер имеет два устойчивых состояния, одно из которых соответствует двоичной единице, а другое – двоичному нулю.

Термин триггер происходит от английского слова *trigger* – защелка, спусковой крючок. Для обозначения этой схемы в английском языке чаще употребляется термин «flip-flop», что в переводе означает «хлопанье». Это звукоподражательное название электронной схемы указывает на ее способность почти мгновенно переходить («перебрасываться») из одного электрического состояния в другое и наоборот. Поскольку один триггер может запомнить только один разряд двоичного кода, то для запоминания байта нужно 8 триггеров, для запоминания килобайта, соответственно,  $8 \times 2^{10} = 8192$  триггеров. Современные микросхемы памяти содержат миллиарды триггеров.

## **8.2. Алгебра логики как наука об общих операциях над логическими высказываниями**

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

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

Так, например, предложение «Меню в программе – это список возможных вариантов» следует считать высказыванием, так как оно истинное. Предложение «Город Джакарта – столица Микронезии» тоже высказывание, так как оно ложное.

Разумеется, не всякое предложение является логическим высказыванием. Высказываниями не являются, например, предложения «Какого цвета этот дом?» и «Эта тема скучна». Первое предложение вопросительное, а не повествовательное. Второе использует слишком неопределенное понятие «тема скучна».

Предложения типа «Рикки Мартин – самый популярный певец» не являются высказываниями, так как для выяснения его истинности или ложности нужны дополнительные сведения: о каком конкретно регионе, или о какой группе людей идет речь. Такие предложения называются высказывательными формами.

Высказывательная форма – это повествовательное предложение, которое прямо или косвенно содержит хотя бы одну переменную и становится высказыванием, когда все переменные замещаются своими значениями.

Алгебра логики рассматривает любое высказывание только с одной точки зрения – является ли оно истинным или ложным. Часто трудно установить истинность высказывания. Так, например, высказывание «Численность жителей города Красноярска составляет 961 тыс. чел» может быть одновременно истинным и ложным. Ложным – так как указанное значение неточное и вообще не является постоянным. Истинным – если рассматривать его как некоторое приближение, приемлемое на практике.

Употребляемые в обычной речи слова и словосочетания «не», «и», «или», «если..., то», «тогда и только тогда» и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются логическими связками.

Высказывания, образованные из других высказываний с помощью логических связок, называются составными.

Высказывания, не являющиеся составными, называются элементарными.

Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности элементарных высказываний.

Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через  $A$  обозначено высказывание «Зимой люди катаются на коньках», а через  $B$  – высказывание «Зимой люди катаются на лыжах».

Тогда составное высказывание «Зимой люди катаются на коньках и на лыжах» можно кратко записать как  $A$  и  $B$ . Здесь «и» – логическая связка,  $A$ ,  $B$  – логические переменные, которые могут принимать только два значения – «истина» или «ложь», обозначаемые, соответственно, «1» и «0».

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение. Базовыми являются пять логических операций: отрицание, конъюнкция, дизъюнкция, импликация и эквиваленция.

Каждую логическую операцию можно иллюстрировать таблицей истинности.

**Таблица истинности** – это табличное представление логической операции, в котором перечислены все возможные сочетания значений истинности входных операндов вместе со значением истинности выходного результата операции для каждого из этих сочетаний.

**НЕ.** Операция, выражаемая словом «НЕ», называется **отрицанием** и обозначается чертой над высказыванием (или знаком  $\neg$ ).

Высказывание  $\overline{A}$  истинно, когда А ложно, и ложно, когда А истинно.

*Пример:* «Луна – спутник Земли» (А); «Луна – не спутник Земли» ( $\overline{A}$ ).

Таблица истинности логической операции «не» приведена в таблице 8.1.

Таблица 8.1 – Таблица истинности логического отрицания «НЕ»

x	$\overline{x}$
0	1
1	0

**И.** Операция, выражаемая связкой «И», называется **конъюнкцией** (лат. *conjunctio* – соединение) или логическим умножением и обозначается точкой «·» (может также обозначаться знаками  $\wedge$  или \*).

Высказывание А и В истинно тогда и только тогда, когда оба высказывания А и В истинны.

Например, высказывание «10 делится на 2 и 5 больше 3» истинно, а высказывания «10 делится на 2 и 5 не больше 3» – ложны. Таблица истинности логической операции «и» приведена в таблице 8.2.

Таблица 8.2 – Таблица истинности логического умножения (конъюнкции) «И»

X	Y	$X \wedge Y$
0	0	0
0	1	0
1	0	0
1	1	1

**ИЛИ.** Операция, выражаемая связкой «ИЛИ» (в неисключающем смысле этого слова), называется **дизъюнкцией** (лат. *disjunctio* – разделение) или логическим сложением и обозначается знаком  $\vee$  (или плюсом +).

Высказывание  $A \vee B$  ложно тогда и только тогда, когда оба высказывания  $A$  и  $B$  ложны.

Например, высказывания «10 не делится на 2» или «5 не больше 3» ложны, а высказывания «10 делится на 2» или «5 больше 3», – истинны. Таблица истинности логической операции «или» приведена в таблице 8.3.

Таблица 8.3 – Таблица истинности логического сложения (дизъюнкции) «ИЛИ»

X	Y	$X \vee Y$
0	0	0
0	1	1
1	0	1
1	1	1

**ЕСЛИ–ТО.** Операция, выражаемая связками «если ..., то», «из ... следует», «... влечет ...», называется **импликацией** (лат. *implico* – тесно связаны) и обозначается знаком  $\rightarrow$  или  $\Rightarrow$ .

Высказывание  $A \rightarrow B$  ложно тогда и только тогда, когда  $A$  истинно, а  $B$  ложно.

Таблица истинности логической операции «импликация» приведена в таблице 8.4.

Таблица 8.4 – Таблица истинности импликации «ЕСЛИ–ТО»

x	y	$x \rightarrow y$
0	0	1
0	1	1
1	0	0
1	1	1

В обычной речи связка «если..., то» описывает причинно-следственную связь между высказываниями. Но в логических операциях смысл высказываний не учитывается. Рассматривается только их истинность или ложность.

**РАВНОСИЛЬНО.** Операция, выражаемая связками «тогда и только тогда», «необходимо и достаточно», «... равносильно ...», называется **эквиваленцией** или двойной импликацией и обозначается знаком  $\Leftrightarrow$  или  $\sim$  или  $\Leftrightarrow$ .

Высказывание  $A \Leftrightarrow B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают.



Например, высказывание «24 делится на 6 тогда и только тогда, когда 24 делится на 3 – истинно, а высказывание «24 делится на 6 тогда и только тогда, когда 24 делится на 5» – ложно.

Таблица истинности логической операции «эквиваленции» приведена в таблице 8.5.

Таблица 8.5 – Таблица истинности эквиваленция, связка «Тогда и только тогда»

x	y	$x \leftrightarrow y$
0	0	1
0	1	0
1	0	0
1	1	1

Импликацию можно выразить через дизъюнкцию и отрицание:

$$A \rightarrow B = \bar{A} \vee B.$$

Эквиваленцию можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \leftrightarrow B = (\bar{A} \vee B) \cdot (\bar{B} \vee A).$$

Таким образом, операций отрицания, дизъюнкции и конъюнкции достаточно, чтобы описывать и обрабатывать логические высказывания.

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

### 8.3. Понятие логической формулы. Законы алгебры логики

Логические формулы. С помощью логических переменных и символов логических операций любое высказывание можно формализовать, то есть заменить логической формулой.

Всякая логическая переменная и символы «истина» («1») и «ложь» («0») – формулы.

Если A и B – формулы, то  $\bar{A}$ ,  $A \cdot B$ ,  $A \vee B$ ,  $A \rightarrow B$ ,  $A \leftrightarrow B$  – формулы.

Никаких других формул в алгебре логики нет.

В качестве примера рассмотрим высказывание «Если я куплю яблоки или абрикосы, то приготовлю фруктовый пирог».

Это высказывание формализуется в виде  $(A \vee B) \rightarrow C$ .

Как показывает анализ формулы  $(A \vee B) \rightarrow C$ , при определенных сочетаниях значений переменных  $A$ ,  $B$  и  $C$  она принимает значение «истина», а при некоторых других сочетаниях – значение «ложь» (разберите самостоятельно эти случаи). Такие формулы называются выполнимыми.

Некоторые формулы принимают значение «истина» при любых значениях истинности входящих в них переменных. Таковой будет, например, формула  $A \vee \bar{A}$ , соответствующая высказыванию «Этот треугольник прямоугольный или косоугольный». Эта формула истинна и тогда, когда треугольник прямоугольный, и тогда, когда треугольник не прямоугольный.

Такие формулы называются тождественно истинными формулами или **тавтологиями**.

Высказывания, которые формализуются тавтологиями, называются **логически истинными высказываниями**.

В качестве другого примера рассмотрим формулу  $A * \bar{A}$ , которой соответствует, например, высказывание «Катя самая высокая девочка в классе, и в классе есть девочки выше Кати». Очевидно, что эта формула ложна, так как либо  $A$ , либо  $\bar{A}$  обязательно ложно.

Такие формулы называются тождественно ложными формулами или **противоречиями**.

Высказывания, которые формализуются противоречиями, называются **логически ложными высказываниями**.

Если две формулы  $A$  и  $B$  одновременно, т. е. при одинаковых наборах значений входящих в них переменных, принимают одинаковые значения, то они называются **равносильными**.

Равносильность двух формул алгебры логики обозначается символом « $\Leftrightarrow$ » или символом « $\equiv$ ». Замена формулы другой, ей равносильной, называется равносильным преобразованием данной формулы.

### ***Основные законы алгебры логики***

В алгебре логики выполняются следующие основные законы, позволяющие производить тождественные преобразования логических выражений (табл. 8.6):

Таблица 8.6 – Основные законы алгебры логики

Закон	Для ИЛИ	Для И
Переместительный	$x \vee y = y \vee x$	$x \cdot y = y \cdot x$
Сочетательный	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Распределительный	$x \cdot (y \vee z) = x \cdot y \vee x \cdot z$	$x \vee (y \cdot z) = (x \vee y) \cdot (x \vee z)$
Правила де Моргана	$\overline{x \vee y} = \bar{x} \cdot \bar{y}$	$\overline{x \cdot y} = \bar{x} \vee \bar{y}$
Идемпотенции	$x \vee x = x$	$x \cdot x = x$
Поглощения	$x \vee (x \cdot y) = x$	$x \cdot (x \vee y) = x$
Склеивания	$(x \cdot y) \vee (\bar{x} \cdot y) = y$	$(x \vee y) \cdot (\bar{x} \vee y) = y$
Операция переменной с ее инверсией	$x \vee \bar{x} = 1$	$x \cdot \bar{x} = 0$
Операция с константами	$x \vee 0 = x; x \vee 1 = 1$	$x \cdot 1 = x; x \cdot 0 = 0$
Двойного отрицания	$\overline{\bar{x}} = x$	

#### 8.4. Таблицы истинности

Таблица истинности логической формулы выражает соответствие между всевозможными наборами значений переменных и значениями формулы.

Для формулы, которая содержит две переменные, таких наборов значений переменных всего  $2^2$  – четыре:

(0, 0), (0, 1), (1, 0), (1, 1).

Если формула содержит три переменные, то возможных наборов значений переменных  $2^3$  – восемь:

(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1).

Количество наборов для формулы с четырьмя переменными равно шестнадцати и т.д.

Удобной формой записи при нахождении значений формулы является таблица, содержащая кроме значений переменных и значений формулы также и значения промежуточных формул.

##### Пример 1

Составьте таблицу истинности для формулы  $\bar{x} \cdot y \vee \overline{x \vee y} \vee x$ , которая содержит две переменные  $x$  и  $y$ . В двух первых столбцах таблицы запишите четыре возможных пары значений этих переменных, в последующих столбцах – значения промежуточных формул и в последнем столбце – значение формулы. В результате получите таблицу 8.7.

Таблица 8.7 – Таблица истинности для примера 1

Переменная		Промежуточная логическая формула					Формула
$x$	$y$	$\bar{x}$	$\bar{x} \cdot y$	$x \vee y$	$\overline{x \vee y}$	$x \cdot y \vee \overline{x \vee y}$	$\overline{x \cdot y \vee \overline{x \vee y \vee x}}$
0	0	1	0	0	1	1	1
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	1	0	0	1	0	0	1

Из таблицы видно, что при всех наборах значений переменных  $x$  и  $y$  формула  $\overline{x \cdot y \vee \overline{x \vee y \vee x}}$  принимает значение 1, т. е. является тождественно истинной.

*Пример 2*

Таблица истинности для формулы  $\overline{x \vee y} \cdot (x \cdot \bar{y})$  приведена в таблице 8.8.

Таблица 8.8 – Таблица истинности для примера 2

Переменная		Промежуточная логическая формула				Формула
$x$	$y$	$x \vee y$	$\overline{x \vee y}$	$\bar{y}$	$x \cdot \bar{y}$	$\overline{x \vee y} \cdot (x \cdot \bar{y})$
0	0	0	1	1	0	0
0	1	1	0	0	0	0
1	0	1	0	1	1	0
1	1	1	0	0	0	0

Из таблицы видно, что при всех наборах значений переменных  $x$  и  $y$  формула  $\overline{x \vee y} \cdot (x \cdot \bar{y})$  принимает значение 0, т. е. является тождественно ложной.

*Пример 3*

Таблица истинности для формулы  $\overline{x \vee \bar{y} \vee \overline{x \cdot z}}$  приведена в таблице 8.9.

Таблица 8.9 – Таблица истинности для примера 3

Переменная			Промежуточная логическая формула					Формула
$x$	$y$	$z$	$\bar{y}$	$x \vee \bar{y}$	$\overline{x \vee y}$	$\bar{x}$	$\bar{x} \cdot z$	$\overline{x \vee y \vee \bar{x} \cdot z}$
0	0	0	1	1	0	1	0	0
0	0	1	1	1	0	1	1	1
0	1	0	0	0	1	1	0	1
0	1	1	0	0	1	1	1	1
1	0	0	1	1	0	0	0	0
1	0	1	1	1	0	0	0	0
1	1	0	0	1	0	0	0	0
1	1	1	0	1	0	0	0	0

Из таблицы видно, что формула  $\overline{x \vee y \vee \bar{x} \cdot z}$  в некоторых случаях принимает значение 1, а в некоторых – 0, т. е. является выполнимой.

### 8.5. Системы логических элементов

Системой логических элементов называется функционально полный набор логических элементов, объединенных общими электрическими, конструктивными и технологическими параметрами и использующих одинаковый тип межэлементных связей.

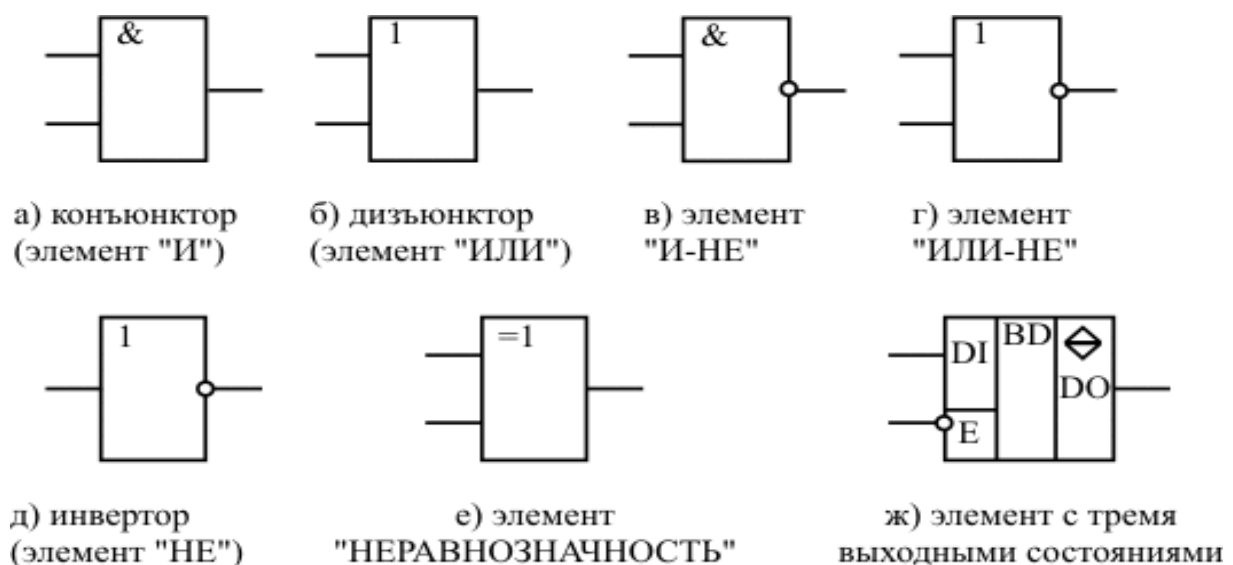


Рисунок 8.3 – Условно-графические обозначения логических элементов

Системы элементов содержат элементы для выполнения логических операций, запоминающие элементы, элементы, реализующие функции узлов ЭВМ, а также элементы для усиления, восстановления и формирования сигналов стандартной формы.

**Условно-графические обозначения (УГО)** некоторых логических элементов представлены на рис. 8.3.

УГО элемента представляет собой прямоугольник, к которому слева подходят входные сигналы, а справа выходят выходные. Внутри прямоугольника ставится условное обозначение выполняемой элементом логической функции. Если значение выходного сигнала принимает инверсное значение по отношению к обозначенной внутри элемента функции, то данный выход обозначается на УГО элемента кружком (рис. 8.3, в ...8.3, д). Аналогично, если активным уровнем входного сигнала является логический «0», то данный вход обозначается кружком (вход Е элемента 8.3, ж).

Таблицы истинности для УГО инвертера (элемента «НЕ»), конъюнктора (элемента «И»), дизъюнктора (элемента «ИЛИ») приведены в параграфе 8.2 в таблицах 8.1, 8.2, 8.3 соответственно.

Элемент «И–НЕ» состоит из элемента И и инвертера и осуществляет отрицание результата схемы И.

Связь между выходом  $z$  и входами  $x$  и  $y$  схемы записывают следующим образом:  $z = \overline{x \cdot y}$ , где  $\overline{x \cdot y}$  читается как «инверсия  $x$  и  $y$ ».

Для элемента «И–НЕ» (рис. 8.3, в) возможные состояния описываются таблицей 8.10.

Таблица 8.10 – Таблица истинности «И–НЕ»

x	y	И–НЕ
0	0	1
0	1	1
1	0	1
1	1	0

Элемент «ИЛИ–НЕ» состоит из элемента ИЛИ и инвертера и осуществляет отрицание результата схемы ИЛИ.

Связь между выходом  $z$  и входами  $x$  и  $y$  схемы записывают следующим образом:  $z = \overline{x \vee y}$ , где  $\overline{x \vee y}$  читается как «инверсия  $x$  или  $y$ ».

Для элемента «ИЛИ–НЕ» (рис. 8.3, г) возможные состояния описываются таблицей 8.11.

Таблица 8.11 – Таблица истинности «ИЛИ–НЕ»

x	y	ИЛИ–НЕ
0	0	1
0	1	0
1	0	0
1	1	0

### Вопросы для повторения и самоконтроля

1. Кто разработал формальную логику и почему она так называется?
2. Какую идею высказал Г. В. Лейбниц о рассуждениях?
3. Кто является основателем математической логики?
4. Кто из наших современников сумел ликвидировать разрыв между алгебраической теорией логики и ее практическим применением?
5. Почему математический аппарат алгебры логики удобен для описания того, как функционируют аппаратные средства компьютера?
6. Для чего служит электронная схема «вентиль»?
7. Что такое триггер?
8. Сформулируйте понятие алгебры логики.
9. Что такое логическое высказывание?
10. Что такое высказывательная форма логического высказывания?
11. Какие слова относятся к логическим связкам (операциям)?
12. Что такое составное и элементарное высказывания?
13. Охарактеризуйте понятие «таблица истинности».
14. Логическое отрицание – обозначение и таблица истинности.
15. Логическое умножение – обозначение и таблица истинности.
16. Логическое сложение – обозначение и таблица истинности.
17. Импликация – обозначение и таблица истинности.
18. Эквиваленция – обозначение и таблица истинности.
19. Через какие другие логические функции можно выразить импликацию и эквиваленцию?
20. Что понимается под логической формулой?
21. Что такое выполняемая логическая формула?
22. Что такое тождественно истинная логическая формула?
23. Что такое тождественно ложная логическая формула?
24. Что такое равносильная логическая формула?

25. Основные законы алгебры логики.
26. Какое количество наборов значений переменных необходимо написать в таблице истинности для  $N$  переменных?
27. Как решаются таблицы истинности?

### Задания для самостоятельной работы

1. Установите, какие из следующих предложений являются логическими высказываниями, а какие – нет (объясните почему):

- а) «Солнце есть спутник Земли»;
- б) « $2+3=4$ »;
- в) «Сегодня отличная погода»;
- г) «В романе Л.Н. Толстого «Война и мир» 3 432 536 слов»;
- д) «Санкт-Петербург расположен на Неве»;
- е) «Музыка Баха слишком сложна»;
- ж) «Первая космическая скорость равна 7.8 км/сек»;
- з) «Железо – металл»;
- и) «Если один угол в треугольнике прямой, то треугольник будет тупоугольным»;
- к) «Если сумма квадратов двух сторон треугольника равна квадрату третьей, то он прямоугольный».

2. Укажите, какие из высказываний предыдущего упражнения истинны, какие – ложны, а какие относятся к числу тех, истинность которых трудно или невозможно установить.

3. Приведите примеры истинных и ложных высказываний:

- а) из арифметики;
- б) физики;
- в) биологии;
- г) информатики;
- д) геометрии;
- е) жизни.

4. Сформулируйте отрицания следующих высказываний или высказывательных форм:

- а) «Эльбрус – высочайшая горная вершина Европы»;
- б) « $2 > 5$ »;
- в) « $10 < 7$ »;
- г) «Все натуральные числа целые»;
- д) «Через любые три точки на плоскости можно провести окружность»;



- е) «Теннисист Кафельников не проиграл финальную игру»;
- ж) «Мишень поражена первым выстрелом»;
- з) «Это утро ясное и теплое»;
- и) «Число  $n$  делится на 2 или на 3»;
- к) «Этот треугольник равнобедренный и прямоугольный»;
- л) «На контрольной работе каждый ученик писал своей ручкой».

5. Определите, какие из высказываний (высказывательных форм) в следующих парах являются отрицаниями друг друга, а какие нет:

- а) « $5 < 10$ », « $5 > 10$ »;
  - б) « $10 > 9$ », « $10 \leq 9$ »;
  - в) «Мишень поражена первым выстрелом», «Мишень поражена вторым выстрелом»;
  - г) «Машина останавливалась у каждого из двух светофоров», «Машина не останавливалась у каждого из двух светофоров»;
  - д) «Человечеству известны все планеты Солнечной системы», «в Солнечной системе есть планеты, неизвестные человечеству»;
  - е) «Существуют белые слоны», «Все слоны серые»;
  - ж) «Кит – млекопитающее», «Кит – рыба»;
  - з) «Неверно, что точка  $A$  не лежит на прямой  $a$ », «Точка  $A$  лежит на прямой  $a$ »;
  - и) «Прямая  $a$  параллельна прямой  $b$ », «Прямая  $a$  перпендикулярна прямой  $b$ »;
  - к) «Этот треугольник равнобедренный и прямоугольный», «Этот треугольник не равнобедренный или он не прямоугольный».
6. Определите значения истинности высказываний:
- а) «Наличия аттестата о среднем образовании достаточно для поступления в институт»;
  - б) «Наличие аттестата о среднем образовании необходимо для поступления в институт»;
  - в) «Если целое число делится на 6, то оно делится на 3»;
  - г) «Подобие треугольников является необходимым условием их равенства»;
  - д) «Подобие треугольников является необходимым и достаточным условием их равенства»;
  - е) «Треугольники подобны только в случае их равенства»;
  - ж) «Треугольники равны только в случае их подобия»;

з) «Равенство треугольников является достаточным условием их подобия»;

и) «Для того, чтобы треугольники были неравны, достаточно, чтобы они были неподобны»;

к) «Для того, чтобы четырехугольник был квадратом, достаточно, чтобы его диагонали были равны и перпендикулярны».

7. Подставьте в приведенные ниже высказывательные формы вместо логических переменных  $a, b, c, d$  такие высказывания, чтобы полученные таким образом составные высказывания имели смысл в повседневной жизни:

а) если ( $a$  или ( $b$  и  $c$ )), то  $d$ ;

б) если ( $\text{не } a$  и  $\text{не } b$ ), то ( $c$  или  $d$ );

в) ( $a$  или  $b$ ) тогда и только тогда, когда ( $c$  и  $\text{не } d$ ).

8. Формализуйте следующий вывод: «Если  $a$  и  $b$  истинны, то  $c$  – истинно. Но  $c$  – ложно: значит,  $a$  или  $b$  ложны».

9. Формализуйте предостережение, которое одна жительница древних Афин сделала своему сыну, собиравшемуся заняться политической деятельностью: «Если ты будешь говорить правду, то тебя возненавидят люди. Если ты будешь лгать, то тебя возненавидят боги. Но ты должен говорить правду или лгать. Значит, тебя возненавидят люди или возненавидят боги».

Формализуйте также ответ сына: «Если я буду говорить правду, то боги будут любить меня. Если я буду лгать, то люди будут любить меня. Но я должен говорить правду или лгать. Значит, меня будут любить боги или меня будут любить люди».

10. Пусть  $a = \text{«это утро ясное»}$ ,  $b = \text{«это утро теплое»}$ . Выразите следующие формулы на обычном языке:

а)  $a \cdot \bar{b}$  з)  $\overline{a \vee b}$  ж)  $\overline{a \cdot b}$  к)  $\overline{a \rightarrow b}$

б)  $\overline{a \cdot b}$  д)  $a \vee \bar{b}$  з)  $\overline{a \vee b}$  л)  $\overline{a \rightarrow b}$

в)  $\overline{a \cdot \bar{b}}$  е)  $\overline{a \vee \bar{b}}$  и)  $\overline{a \cdot b}$  м)  $\overline{a \rightarrow b}$

11. Из трех данных высказываний  $a, b, c$  постройте составное высказывание, которое истинно, когда истинно какое-либо одно из данных высказываний, и только в этом случае.

12. Определите с помощью таблиц истинности, какие из следующих формул являются тождественно истинными или тождественно ложными:

- а)  $\overline{\overline{a} \vee b} \cdot (a \cdot b \vee b)$ ;      д)  $a \cdot (b \cdot (\overline{a} \vee \overline{b}))$ ;  
 б)  $((a \vee \overline{b}) \rightarrow b) \cdot (\overline{a} \vee b)$ ;      е)  $\overline{(\overline{a} \vee b) \cdot (\overline{b} \vee c) \vee \overline{a} \vee c}$ ;  
 в)  $\overline{a \cdot b} \Leftrightarrow (\overline{a} \vee \overline{b})$ ;      ж)  $\overline{(a \rightarrow b) \Leftrightarrow (\overline{b} \rightarrow \overline{a})}$ .  
 г)  $a \cdot b \cdot (c \vee \overline{e} \vee d) \cdot \overline{b}$ ;

13. Определите с помощью таблиц истинности, какие из следующих формул являются тождественно истинными или тождественно ложными:

- а)  $\overline{a \cdot \overline{c} \vee c \cdot (b \vee \overline{c})} \vee (a \vee \overline{b}) \cdot c$ ;  
 б)  $a \cdot (b \vee \overline{c}) \vee \overline{a} \cdot b$ ;  
 в)  $(\overline{a} \vee c) \cdot a \cdot \overline{c} \cdot (b \vee \overline{c}) \cdot \overline{b} \cdot c$ ;  
 д)  $a \cdot b \cdot c \vee a \cdot \overline{b} \cdot c \vee a \cdot b \cdot \overline{c} \cdot d$ .

### *Ответы к заданиям для самостоятельной работы*

1. Являются высказываниями: а), г), д), ж), з), и), к); не являются высказываниями: б); в); е).

2. Истинные: д), з), к); ложные: а), и); истинность трудно установить: г); можно рассматривать и как истинное, и как ложное в зависимости от требуемой точности представления: ж).

3. Например, истинные высказывания:

а) « $2 + 2 = 4$ »; б) «Сила притяжения тел обратно пропорциональна квадрату расстояния между ними»; в) «зайцы питаются растениями»; г) «бит – фундаментальная единица информации, используемая в теории информации»; д) «два треугольника равны, если две стороны и угол между ними одного треугольника равны двум сторонам и углу между ними другого треугольника»; е) «понедельник – первый день недели».

Ложные высказывания: а) « $4 + 3 = 5$ »; б) «тело падает на Землю с ускорением, пропорциональным своей массе»; в) «животные это неживая природа»; г) «информатика – наука о термической обработке металлов»; д) «квадрат это фигура, у которой пять сторон»; е) «лев – домашнее животное».

4. а) «Эльбрус – не высочайшая горная вершина Европы»; б) « $2 < 5$ »; в) « $10 >= 7$ »; г) «не все натуральные числа целые»; д) «не через любые три точки на плоскости можно провести окруж-

ность»; е) «теннисист Кафельников проиграл финальную игру»; ж) «мишень не поражена первым выстрелом»; з) «это утро не ясное или оно не теплое» (Пояснение. Пусть  $A =$  «это утро ясное», а  $B =$  «это утро теплое». Тогда «это утро ясное и теплое» можно записать как  $A \cdot B$ , отрицанием чего является  $\overline{A \cdot B} = \overline{A} \vee \overline{B}$ , что соответствует высказывательной форме «это утро не ясное или оно не не теплое»; и) «число  $n$  не делится на 2 и оно делится на 3»; к) «этот треугольник не равнобедренный или он не прямоугольный»; л) «не каждый ученик писал контрольную своей ручкой» (вариант: «кто-то писал контрольную не своей ручкой»).

5. Являются отрицаниями друг друга: б), г), д), к); не являются отрицаниями друг друга: а), в), е), ж), з), и).

6. Истинны: б), в), г), з), к), и); ложны: а), д), е), ж).

$$8. ((a \cdot b \rightarrow c) \cdot \bar{c}) \rightarrow (\bar{a} \vee \bar{b}).$$

9. Решение. Введем обозначения для логических высказываний:  $a$  – «ты будешь говорить правду»;  $b$  – «тебя возненавидят люди»;  $c$  – «тебя возненавидят боги». Договоримся считать, что некоторое заданное высказывание  $x$  истинно, если нет оговорки. Тогда предостережение матери можно записать так:

$$(((a \rightarrow b) \vee (a \rightarrow c)) \cdot (a \rightarrow \bar{a})) \rightarrow (b \vee c).$$

А ответ сына – так:  $(((a \rightarrow \bar{c}) \vee (\bar{a} \rightarrow \bar{b})) \cdot (a \rightarrow \bar{a})) \rightarrow (\bar{b} \vee \bar{c}).$

10.

- а) «Это утро ясное и теплое»;
- б) «Это утро ясное и оно не теплое»;
- в) «Это утро не ясное и оно не теплое»;
- г) «Это утро не ясное или оно теплое»;
- д) «Это утро ясное или оно не теплое»;
- е) «Это утро не ясное или оно не теплое»;
- ж) «Это утро не ясное или не теплое»;
- з) «Это утро не ясное и не теплое»;
- и) «Это утро ясное или не теплое»;
- к) «Если это утро ясное, то оно не теплое»;
- л) «Если это утро не ясное, то оно теплое»;
- м) «Это утро ясное и не теплое».

11.  $a \cdot \bar{b} \cdot \bar{c} \vee \bar{a} \cdot b \cdot \bar{c} \vee \bar{a} \cdot \bar{b} \cdot c$

12. Тождественно истинные: а), в), е); тождественно ложные: г), д), ж).

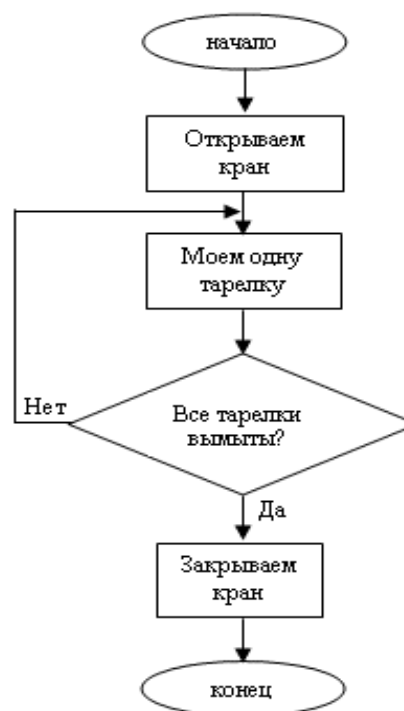
13. Тождественно истинные: а), в), е); тождественно ложные: г), д).

# ГЛАВА 9. АЛГОРИТМЫ. АЛГОРИТМИЗАЦИЯ. АЛГОРИТМИЧЕСКИЕ ЯЗЫКИ

## 9.1. Алгоритм. Свойства алгоритмов

### *Сущность алгоритма*

Понятие алгоритма такое же основополагающее для информатики, как и понятие информации. Именно поэтому важно в нем разобраться, что и делает группа программистов на рисунке 9.1.



*Рисунок 9.1 – Программисты, изучающие алгоритм*

Название «алгоритм» произошло от имени величайшего среднеазиатского математика Мухаммеда ибн Муса ал-Хорезми (*Alhorithmi*) (рис. 9.2), жившего в 783–850 гг. В своей книге «Об индийском счете» он изложил правила записи натуральных чисел с помощью арабских цифр и правила действий над ними «столбиком», знакомые теперь каждому школьнику. В XII веке эта книга была переведена на латынь и получила широкое распространение в Европе.

Алгоритм – заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

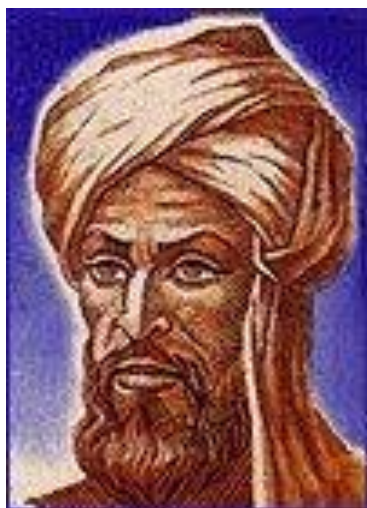


Рисунок 9.2 – Мухаммеда ибн Муса ал-Хорезми

Исполнитель алгоритма – это абстрактная или реальная система, способная выполнить действия, предписываемые алгоритмом. В информатике универсальным исполнителем алгоритмов является компьютер.

#### *Основные свойства алгоритмов*

1. Понятность для исполнителя – исполнитель алгоритма должен понимать, как его выполнять. Иными словами, имея алгоритм и произвольный вариант исходных данных, исполнитель должен знать, как надо действовать для выполнения этого алгоритма.

2. Дискретность (прерывность, отдельность) – алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов (этапов).

3. Определенность – каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

4. Результативность (или конечность) состоит в том, что за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

5. Массовость означает, что алгоритм решения задачи разрабатывается в общем виде, т. е. он должен быть применим для

некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

## 9.2. Формы записи алгоритмов

Существуют множество различных форм записи алгоритмов. Это связано с тем, что каждый исполнитель алгоритмов «понимает» лишь такой алгоритм, который записан на его языке и по его правилам. Условно выделяют 4 формы записи алгоритмов:

1. Словесно-пошаговая (текстовая).
2. Табличная.
3. Запись на алгоритмическом языке.
4. Графическая форма записи (блок-схема).

Предваряя описание алгоритмов, следует рассмотреть команду «присваивание».



Рисунок 9.3 – Оператор присваивания

Величина получает свое значение путем выполнения команды присваивания. Она обозначается знаком  $:=$  (присвоить) и записывается следующим образом: ИМЯ ВЕЛИЧИНЫ  $:=$  ЗНАЧЕНИЕ (рис. 9.3).

Нельзя путать знак присваивания со знаком равенства, так, при выполнении следующей команды присваивания  $Z := Z + 1$ , величина  $Z$  станет равной предыдущему значению  $Z$ , увеличенному на 1.

Ясно, что использование в этой записи знака равенства недопустимо, так как никакое  $Z$  не равно  $Z + 1$ .

После присваивания величине нового значения старое забывается.

### Пример

Определите, чему станет равна величина  $F$  после выполнения следующего ряда присвоений:

$F := 4; F := F * 5; F := 10 + F; F := F / 3; F := F - 10.$

Нам кажется, что в результате будет 0. Проверьте сами



### Словесно-пошаговая (текстовая) форма записи алгоритма

Алгоритм записывается в виде пронумерованных этапов его выполнения.

*Пример:* Алгоритм сложения двух чисел (а и b).

1. Спросить, чему равно число а.
2. Спросить, чему равно число b.
3. Сложить а и b, результат присвоить с.
4. Сообщить результат с.

Словесный способ не имеет широкого распространения, так как такие описания:

- строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

### Табличная форма записи алгоритма

Это запись алгоритма в виде таблицы. Используемые таблицы могут быть различными. Для примера будем использовать упрощенную форму.

*Порядок* составления табличных алгоритмов:

1. Переписать выражение так, как допустимо в информатике.
2. Определить порядок действий.
3. Ввести обозначения промежуточных результатов.
4. Занести полученные действия в таблицу.

Табличная форма записи обычно используется в качестве вспомогательного средства проверки правильности написания алгоритма. Он четко не формализован.

*Пример:* Алгоритм вычисления  $R = 2a + 3b$  приведен в таблице 9.1.

Таблица 9.1 – Алгоритм вычисления

Номер действия	Действие	Величина		Результат
		1	2	
1	*	2	a	k
2	*	3	b	u
3	+	k	u	R

## Запись на алгоритмическом языке<sup>11</sup>

Это запись алгоритма на специальном языке (в том числе и на языке программирования). Она осуществляется по правилам того или иного алгоритмического языка.

Заголовок включает в себя название алгоритма, имена исходных данных (это величины, без которых выполнить алгоритм невозможно) и имена результатов (это величины, значения которых вычисляются в алгоритме).

Для указания начала и конца алгоритма используются служебные слова *нач* и *кон*. Между ними записывают одну или несколько команд алгоритма, их называют *тело* алгоритма.

*Пример:* Алгоритм вычисления значения выражения  $Y = z - a + 2b$ .

<u>алг</u>	ВЗВ	$Y = z - a + 2b$	<- название алгоритма
<u>арг</u>	$z, a, b$		<- исходные данные (аргументы)
<u>рез</u>	$Y$		<- результат
<u>нач</u>			<- начало алгоритма
		$Y := z - a + 2 * b$	<- тело алгоритма
<u>кон</u>			<- конец алгоритма

Четвертую (*графическую*) форму записи алгоритма рассмотрим более подробно.

### 9.3. Графическая форма записи алгоритма

Алгоритм записывают в виде схемы, состоящей из блоков (геометрических фигур) с размещенными в них действиями. Блоки соединяются стрелочками и показывают структуру всего алгоритма. Алгоритм в виде блок-схемы начинается блоком «начало» и заканчивается блоком «конец». Основные блоки приведены на рисунке 9.4.

При составлении блок-схемы алгоритма сначала выделяют исходные данные (все переменные величины после знака равенства и в условии) и результат (величины которые необходимо найти). Если в задании подразумеваются, но не указываются имена величин, то они обозначаются самостоятельно.

При отсутствии исходных данных блок ввода не пишется. В один блок можно поместить одно действие. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий.

---

<sup>11</sup>Подробнее о языках программирования в п. 9.4.

По умолчанию не требуется рисование стрелок при проведении линий слева направо и сверху вниз. Во всех остальных случаях стрелки необходимы.

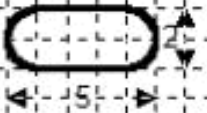
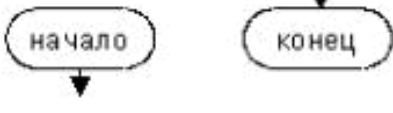
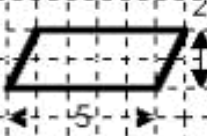
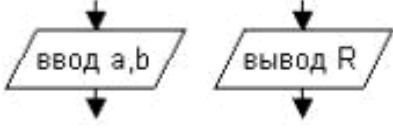
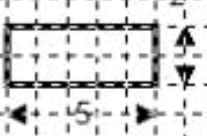


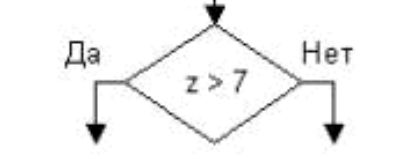

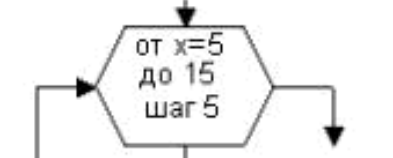
вид блока	название / назначение	примеры записи
	<b>блок начала / конца алгоритма</b> обозначает начало или конец алгоритма	
	<b>блок ввода / вывода</b> служит для ввода исходных данных и вывода результатов	
	<b>блок действия</b> служит для записи команды присваивания	
	<b>блок логического условия</b> служит для организации ветвления в алгоритме	
	<b>блок цикла</b> служит для организации циклов в алгоритме	

Рисунок 9.4 – Изображение основных блоков

*Пример:* Алгоритм вычисления значения выражения  $K = 3b + 6a$  приведен на рисунке 9.5.

Существуют еще формы записи, которые можно отнести к графическим формам представления алгоритмов. Одной из таких форм является построение структурограмм (диаграмм Несси – Шнайдермана, названных так по фамилиям авторов).

Действия в структурограмме располагаются друг под другом. Это позволяет наглядно отслеживать обработку данных в алгоритмах. Все структуры имеют прямоугольную форму. Заполнение их сходно с аналогичными блоками в блок-схемах, но имеются и отличия.

Ввод исх. данных:  $b, a$   
 Вычисление  $R := 3 \cdot b$   
 Вычисление  $M := 6 \cdot a$   
 Вычисление  $K := R + M$   
 Вычисление  $K = 3 \cdot b + 6 \cdot a$   
 Вывод результата:  $K$

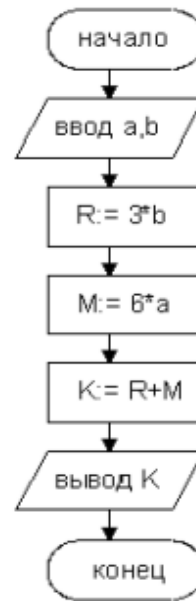


Рисунок 9.5 – Текстовая и блок-схема форма записи алгоритма вычисления выражения  $K = 3b + 6a$

Пример: Рассмотрим блок-схему и структурограмму алгоритма, приведенного на рисунке 9.6.

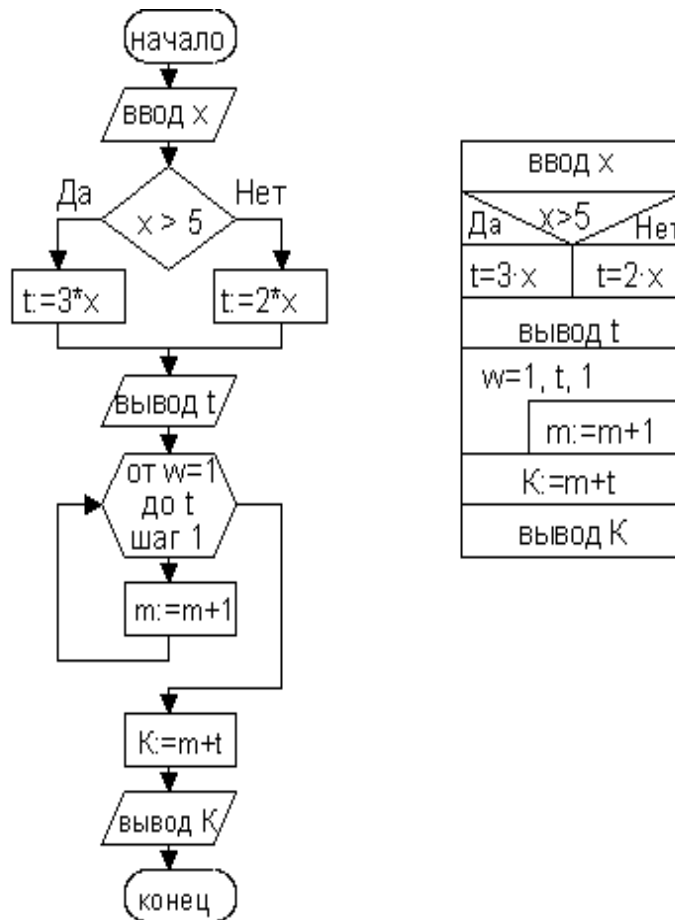


Рисунок 9.6 – Блок-схемное и структурограммное представление алгоритма

Результаты сравнения этих двух форм записи впишите в таблицу 9.2.

Таблица 9.2 – Сравнение представления блок-схемы и структурограммы

Номер	Название блока	Обозначение	
		в блок-схемах	в структурограммах

### 9.3.1. Виды алгоритмов

Выделяют следующие базовые виды алгоритмов: **линейный, разветвляющийся, циклический** (табл. 9.3). Характерной особенностью базовых структур является наличие в них одного входа и одного выхода. При определении вида алгоритма пользуются ключевыми словами.

Таблица 9.3 – Базовые алгоритмические структуры

Вид алгоритма	Ключевые слова	Структура
Алгоритм, в котором есть структура следование, называется линейным. Следование – это расположение действий друг за другом	Ключевых слов нет	<p><b>СЛЕДОВАНИЕ</b></p> <pre> graph TD     Start(( )) --&gt; A[действие1]     A --&gt; B[действие2]     B --&gt; End(( ))         </pre>
Алгоритм, в котором есть структура ветвление, называется разветвляющимся. Ветвление – это выбор действия в зависимости от выполнения какого-нибудь условия	Если... то... иначе...; при... (в значении если...)	<p><b>ВЕТВЛЕНИЕ</b></p> <pre> graph TD     Start(( )) --&gt; C{условие}     C -- Да --&gt; D[действие1]     C -- Нет --&gt; E[действие2]     D --&gt; F(( ))     E --&gt; F     F --&gt; End(( ))         </pre>
Алгоритм, в котором есть структура цикл, называется циклическим. Цикл – это неоднократное повторение каких-либо действий	От...до...; ... раз; пока ...; если... (в значении пока...)	<p><b>ЦИКЛ</b></p> <pre> graph TD     Start(( )) --&gt; G{условие}     G --&gt; H[действие1]     H --&gt; I[действие2]     I --&gt; G     G --&gt; End(( ))         </pre>

Алгоритм, который содержит несколько структур одновременно, называется комбинированным. На рис. 9.7 представлен пример комбинированного алгоритма.

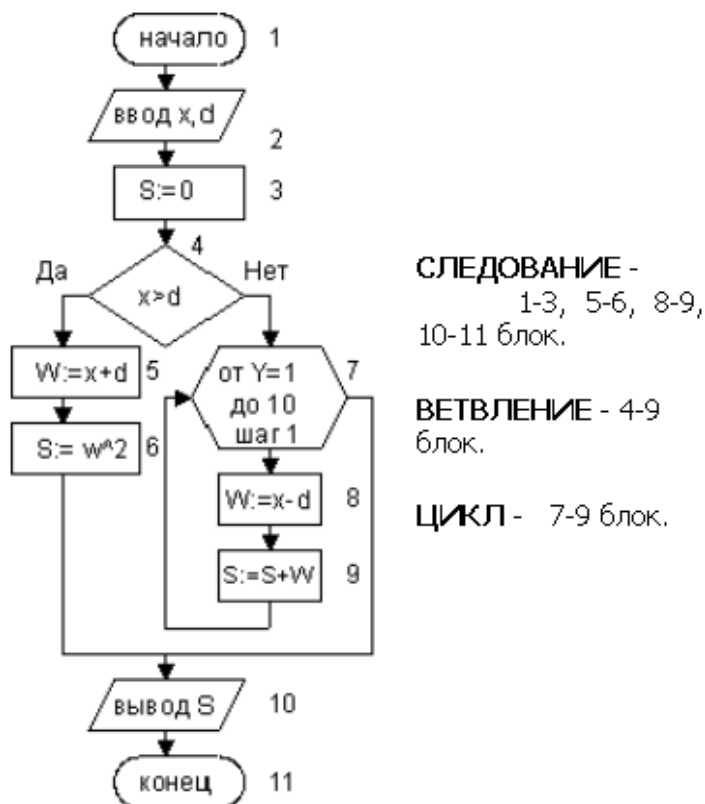


Рисунок 9.7 – Блок-схема комбинированного алгоритма

### Базовая структура «следование»

Образуется последовательностью действий, идущих одно за другим. Структура изображена в таблице 9.3.

### Базовая структура «ветвление»

Ветвление в алгоритмах позволяет выполнить действие (или серию действий) в зависимости от выполнения или невыполнения какого-нибудь условия.

Условие представляет собой строку, содержащую операцию сравнения с использованием знаков  $<>=$ .

Например:  $x > 5$ ;  $s1 < -15.5$ ;  $d\$ = \text{«да»}$ ;  $j <> 4$  (не равно);  $Z3 \geq 3$  (больше или равно);  $t \leq 0$  (меньше или равно).

Если условие соблюдается, то выполняются действия, расположенные в ветви под названием «Да». В случае несоблюдения условия будут выполнены действия, расположенные в ветви «Нет». Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Структура ветвление в разветвляющихся алгоритмах может быть представлена в двух формах: *полной* или *неполной* (рис. 9.8).

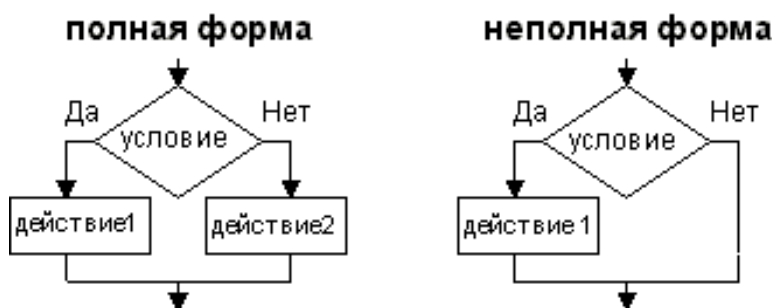


Рисунок 9.8 – Блок-схема алгоритма полного или неполного ветвления

Неполная отличается от полной тем, что в одной из ветвей действия отсутствуют. В таком алгоритме в соответствии с условием либо будут выполнены действия, имеющиеся в ветви, либо начнут сразу выполняться действия, расположенные после ветвления.

На алгоритмическом языке структура «ветвление» записывается следующим образом (табл. 9.4).

Таблица 9.4 – Представление структуры «ветвление» в алгоритмическом языке

В полной форме	В неполной (сокращенной) форме
Если условие то действие 1 иначе действие 2 все	Если условие то действие все
В зависимости от условия в строке <i>если</i> выполняется только одно из действий (или группа действий), расположенных либо в строке <i>то</i> (условие соблюдено), либо в строке <i>иначе</i> (условие не соблюдено)	В этом случае выполнится действие (или группа действий), расположенное в строке <i>то</i> только при соблюдении условия. Если же условие не соблюдается, то исполнитель перейдет к выполнению действий, следующих за служебным словом <i>все</i>

Служебное слово *если* обозначает начало ветвления, а *все* – конец ветвления.

Кроме двух базовых вариантов структуры «ветвление», на практике применяются еще один вариант разветвления «выбор» (рис. 9.9).

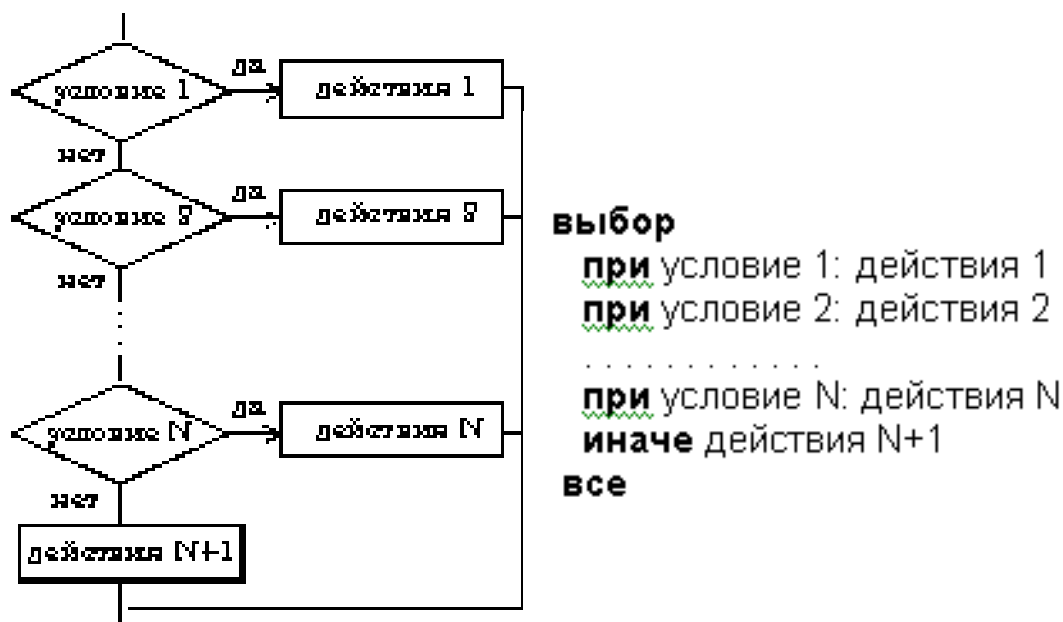


Рисунок 9.9 – Блок-схема алгоритмической структуры «выбор»

Примеры структуры «ветвление» приведены в таблице 9.5.

Таблица 9.5 – Фрагменты блок-схем разветвляющихся алгоритмов

Алгоритмический язык	Язык блок-схем
Если $x > 0$ то $y := \sin(x)$ все	
Если $a > b > 0$ то $a := a * 2; b := 1$ иначе $b := 2 * b$ все	
Выбор при $n = 1$ : $y := \sin(x)$ при $n = 2$ : $y := \cos(x)$ при $n = 3$ : $y := 0$ все	



### Базовая структура «цикл»

Структура «цикл» используется при составлении алгоритмов, в которых необходимо многократно повторять какие-либо действия.

#### Цикл с параметром (цикл «для»)

Для организации цикла с параметром вводится величина (счетчик), которая меняет свое значение от начального до конечного с определенным шагом. Шаг равен разности между следующим и предыдущим значением величины (рис. 9.10).

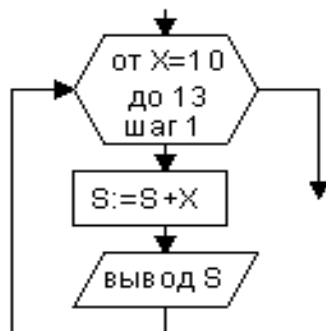


Рисунок 9.10 – Структура цикла с параметром (цикла «для»)

Если при выполнении алгоритма должен получиться ряд ответов, то блок вывода помещается внутри цикла.

#### Пример.

Цикл от  $X = 10$  до 13 шаг 1.  $X$  будет принимать значения равные: 10, 11, 12, 13.

Цикл от  $R = 20$  до 14 шаг 2.  $R$  будет принимать значения равные: 20, 18, 16, 14.

Все действия, размещенные внутри цикла, называются телом цикла. Тело цикла выполняется столько раз, сколько разных значений примет параметр в заданных пределах.

На алгоритмическом языке *начало* и *конец* цикла обозначают служебными словами **нц** и **кц**. Оформление цикла с параметром делается следующим образом (рис. 9.11):

	Например:
условие	<b>для N от 3 до 5 шаг 2</b>
нц	<b>нц</b>
тело цикла	<b>К:=N<sup>2</sup></b>
кц	<b>вывод K</b>
	<b>кц</b>

Рисунок 9.11 – Пример организации цикла с параметром на алгоритмическом языке

### Цикл с логическим условием (цикл «до» )

Для организации цикла также можно использовать блок логического условия.

Тело цикла размещается до проверки условия его окончания. Цикл выполнится хотя бы один раз. Блок-схема и запись на алгоритмическом языке цикла «до» показаны на рисунке 9.12.

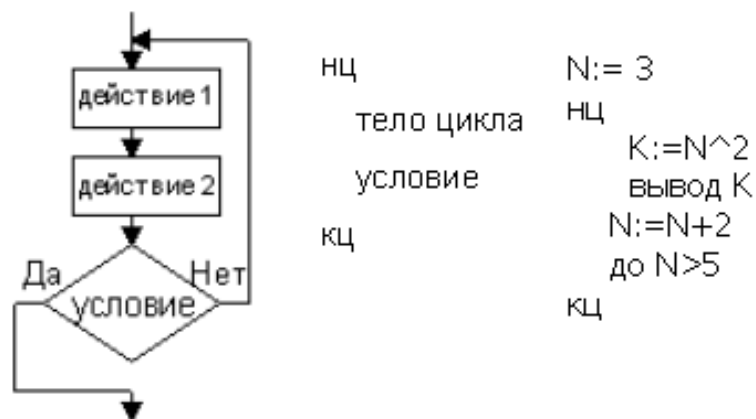


Рисунок 9.12 – Пример организации цикла «до»

### Цикл с логическим условием (цикл «пока»)

Тело цикла размещается после проверки условия его окончания. Цикл может не выполниться ни одного раза. Блок-схема и запись на алгоритмическом языке цикла «пока» показаны на рисунке 9.13. Данный вариант циклической структуры более универсален, так как существует значительное количество задач, где требуется проверка предусловия.

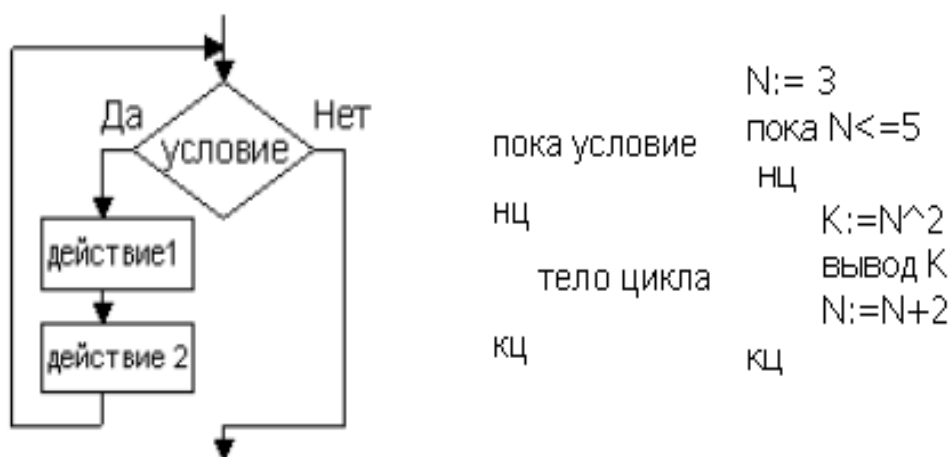
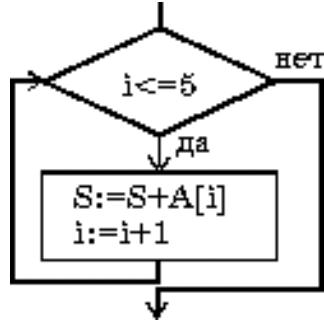
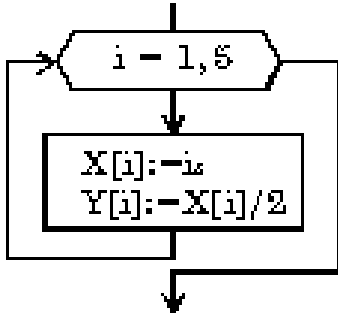


Рисунок 9.13 – Пример организации цикла «пока»

Примеры циклических структур приведены в таблице 9.6.

Таблица 9.6 – Фрагменты блок-схем циклических алгоритмов

Алгоритмический язык	Язык блок-схем
<p>нц пока <math>i \leq 5</math>  <math>S := S + A[i]</math>  <math>i := i + 1</math>  кц</p>	
<p>нц для <math>i</math> от 1 до 5  <math>X[i] := i * i * i</math>  <math>Y[i] := X[i] / 2</math>  кц</p>	

В экономических вычислениях наиболее часто встречаются алгоритмы определения сумм и произведений. Рассмотрим их более подробно.

*Алгоритмы вычисления суммы и произведения*

При вычислении суммы или произведения ряда чисел пользуются соответствующими формулами.

**ФОРМУЛА СУММЫ**  $S_i = S_{i-1} + x_i$

Сумма равна предыдущей сумме плюс аргумент. Начальная сумма равна нулю. При нахождении количества аргумент равен одному.

**ФОРМУЛА ПРОИЗВЕДЕНИЯ**  $P_i = P_{i-1} * x_i$

Произведение равно предыдущему произведению, умноженному на аргумент. Начальное произведение всегда равно единице.

Математически данные формулы записываются как на рисунке 9.14.

Если в аргументе около имени какой-нибудь величины стоит индекс счетчика, то внутри цикла необходимо поставить блок ввода этой величины.



Рисунок 9.14 – Формулы для вычисления суммы и произведения

В качестве примера рассмотрим блок-схемы алгоритмов для приведенных на рисунке 9.15 примеров.

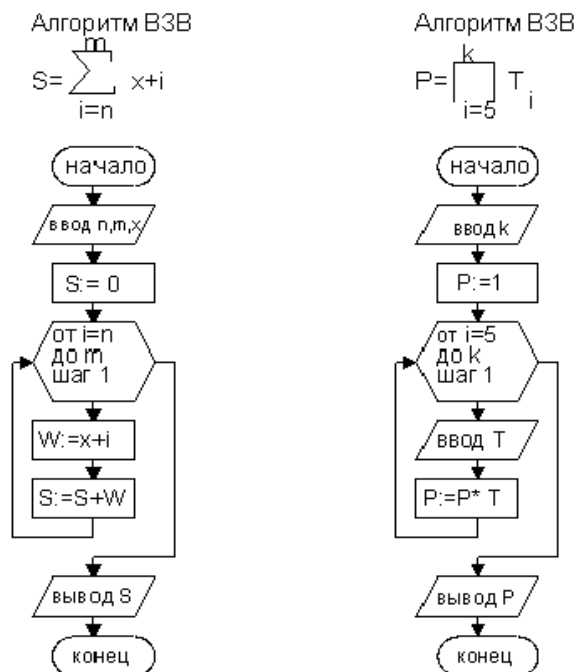


Рисунок 9.15 – Блок-схемы алгоритмов вычисления суммы и произведения

### Вложенные циклы

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

При использовании такой структуры для экономии машинного времени необходимо выносить из внутреннего цикла во внешний все операторы, которые не зависят от параметра внутреннего цикла.

Рассмотрим два примера вычисления вложенных циклов.

*Вложенный цикл «до»*

*Пример*

Вычислите произведение тех элементов заданной матрицы  $A(10, 10)$ , которые расположены на пересечении четных строк и четных столбцов (рис. 9.16).

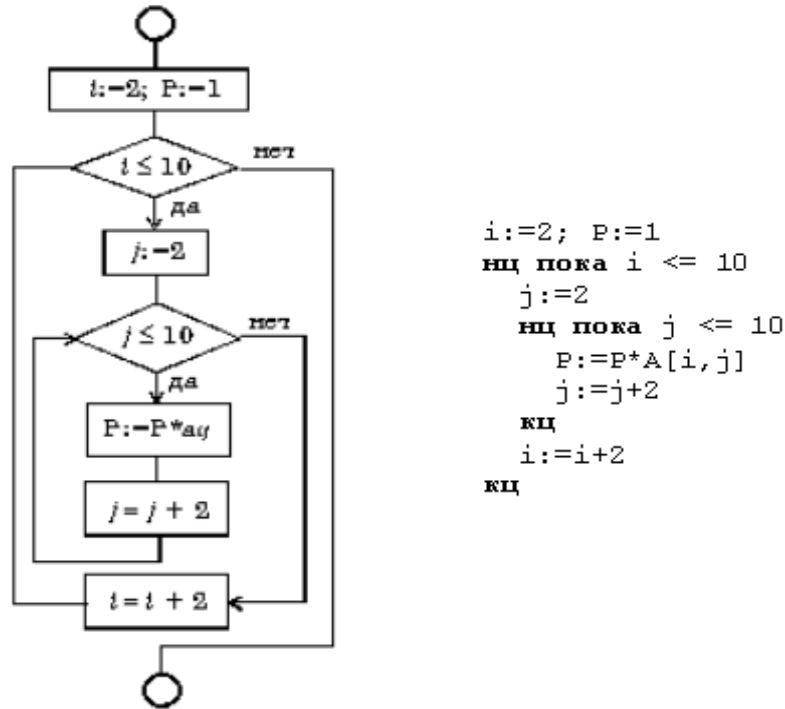


Рисунок 9.16 – Вложенный цикл «до»

*Вложенный цикл «пока»*

*Пример*

Вычислите сумму элементов заданной матрицы  $A(5, 3)$  – рисунок 9.17.

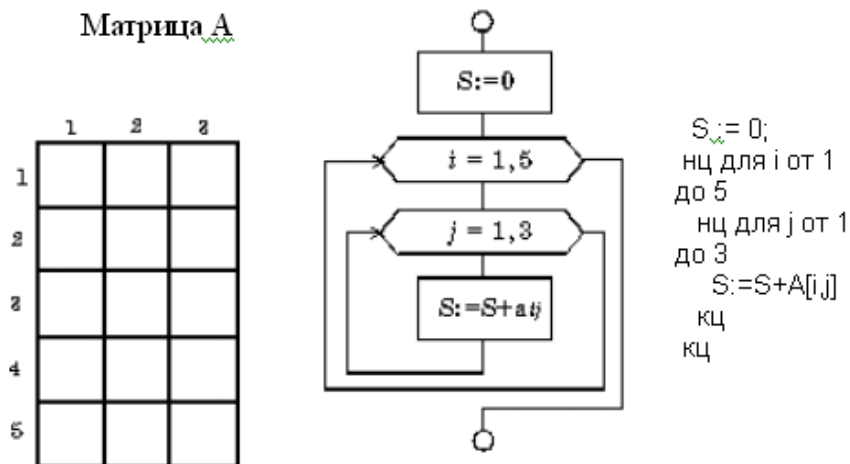


Рисунок 9.17 – Вложенный цикл «пока»

## 9.4. Языки программирования

### 9.4.1. Программный способ записи алгоритмов. Уровни языка программирования

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается некоторый произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.

Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы – компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на понятном ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем.

Следовательно, язык для записи алгоритмов должен быть формализован. Такой язык принято называть языком программирования, а запись алгоритма на этом языке – программой для компьютера.

В настоящее время в мире существует несколько сотен реально используемых языков программирования. Для каждого есть своя область применения.

Любой алгоритм есть последовательность предписаний, выполнив которые можно за конечное число шагов перейти от исходных данных к результату. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования – чем меньше детализация, тем выше уровень языка.

По этому критерию можно выделить следующие уровни языков программирования:

- машинные;
- машинно-ориентированные (ассемблеры);
- машинно-независимые (языки высокого уровня).

**Машинные языки** и **машинно-ориентированные языки** – это языки низкого уровня, требующие указания мелких деталей процесса обработки данных.

Языки же **высокого уровня** имитируют естественные языки, используя некоторые слова разговорного языка и общепринятые математические символы. Эти языки более удобны для человека.

Языки высокого уровня делят:

- на процедурные (алгоритмические) (Basic, Pascal, C и др.), которые предназначены для однозначного описания алгоритмов. Для

решения задачи процедурные языки требуют в той или иной форме явно записать процедуру ее решения;

– логические (Prolog, Lisp и др.), которые ориентированы не на разработку алгоритма решения задачи, а на систематическое и формализованное описание задачи с тем, чтобы решение следовало из составленного описания;

– объектно-ориентированные (Object Pascal, C++, Java и др.), в основе которых лежит понятие объекта, сочетающего в себе данные и действия над ними. Программа на объектно-ориентированном языке, решая некоторую задачу, по сути, описывает часть мира, относящуюся к этой задаче. Описание действительности в форме системы взаимодействующих объектов естественнее, чем в форме взаимодействующих процедур.

При программировании на машинном языке программист может держать под контролем каждую команду и каждую ячейку памяти, использовать все возможности имеющихся машинных операций.

Но процесс написания программы на машинном языке очень трудоемкий и утомительный. Программа получается громоздкой, ее сложно отлаживать, изменять и развивать.

Поэтому в случае, когда нужно иметь эффективную программу, в максимальной степени учитывающую специфику конкретного компьютера, вместо машинных языков используют близкие к ним машинно-ориентированные языки (ассемблеры) или языки высокого уровня.

#### ***9.4.2. Процедурно-ориентированное программирование<sup>12</sup>***

Появление первых электронных вычислительных машин, или компьютеров, ознаменовало новый этап в развитии техники вычислений. Появились специализированные языки программирования, созданные для разработки программ, предназначенных для решения вычислительных задач. Примерами таких языков могут служить FOCAL и FORTRAN.

Основой разработки программ являлась процедурная, или алгоритмическая, организация структуры программного кода. Это было настолько естественно для решения вычислительных задач, что целе-

---

<sup>12</sup>На основе материалов [http://www.intuit.ru/department/pl/javapl/2/javapl\\_2.html](http://www.intuit.ru/department/pl/javapl/2/javapl_2.html) [28]

сообразность такого подхода ни у кого не вызывала сомнений. Исходным в данной методологии было понятие алгоритма.

При увеличении объемов программ для упрощения их разработки появилась необходимость разбивать большие задачи на подзадачи. В языках программирования возникло и закрепилось новое понятие процедуры. Использование процедур позволило разбивать большие задачи на подзадачи и таким образом упростило написание больших программ. Кроме того, процедурный подход позволил уменьшить объем программного кода за счет написания часто используемых кусков кода в виде процедур и их применения в различных частях программы.

Как и алгоритм, процедура представляет собой законченную последовательность действий или операций, направленных на решение отдельной задачи. В языках программирования появилась специальная синтаксическая конструкция, которая также получила название процедуры.

Со временем вычислительные задачи становились все сложнее, а значит, и решающие их программы увеличивались в размерах. Их разработка превратилась в серьезную проблему. Когда программа становится все больше, ее приходится разделять на все более мелкие фрагменты. Основой для такого разбиения как раз и стала процедурная декомпозиция, при которой отдельные части программы, или модули, представляли собой совокупность процедур для решения одной или нескольких задач. Одна из основных особенностей процедурного программирования заключается в том, что оно позволило создавать библиотеки подпрограмм (процедур), которые можно было бы использовать повторно в различных проектах или в рамках одного проекта. При процедурном подходе для визуального представления алгоритма выполнения программы применяется так называемая блок-схема. Соответствующая система графических обозначений была зафиксирована в ГОСТ 19.701-90. Блок-схемный подход к написанию алгоритмов был рассмотрен выше.

Дальнейшее увеличение программных систем способствовало формированию новой точки зрения на процесс разработки программ и написания программных кодов, которая получила название методологии структурного программирования. Ее основой является процедурная декомпозиция предметной области решаемой задачи и организация отдельных модулей в виде совокупности процедур. В рамках этой методологии получило развитие нисходящее проектирование



программ, или проектирование «сверху вниз». Пик популярности идей структурного программирования приходится на конец 70-х – начало 80-х гг.

В этот период основным показателем сложности разработки программ считался ее размер. Общая трудоемкость разработки программ оценивалась специальной единицей измерения – «человеко-месяц», или «человеко-год». А профессионализм программиста напрямую связывался с количеством строк программного кода, который он мог написать и отладить в течение, скажем месяца.

### ***9.4.3. Объектно-ориентированное программирование***

Увеличение размеров программ приводило к необходимости привлечения большего числа программистов, что, в свою очередь, потребовало дополнительных ресурсов для организации их согласованной работы.

Но не менее важными оказались качественные изменения, связанные со смещением акцента использования компьютеров. В эпоху «больших машин» основными потребителями программного обеспечения были крупные заказчики. Стоимость таких вычислительных устройств для небольших предприятий и организаций была слишком высока.

Позже появились персональные компьютеры, которые имели меньшую стоимость и были значительно компактнее. Это позволило широко использовать их в малом и среднем бизнесе. Основными задачами в этой области являются обработка данных и манипулирование ими, поэтому вычислительные и расчетно-алгоритмические задачи с появлением персональных компьютеров отошли на второй план. Как показала практика, традиционные методы процедурного программирования не способны справиться ни с нарастающей сложностью программ и их разработки, ни с необходимостью повышения их надежности. Во второй половине 80-х гг. возникла настоятельная потребность в новой методологии программирования, которая была бы способна решить весь этот комплекс проблем. Ею стало объектно-ориентированное программирование (ООП).

Объектно-ориентированный подход к проектированию основан на представлении предметной области задачи в виде множества моделей для независимой от языка разработки программной системы на основе ее прагматики.

Прагматика определяется целью разработки программной системы, например, обслуживание чемпионата мира по легкой атлетике (который проходил во время написания данного фрагмента). В формулировке цели участвуют предметы и понятия реального мира, имеющие отношение к создаваемой системе (рис. 9.18). При объектно-ориентированном подходе эти предметы и понятия заменяются моделями, т. е. определенными формальными конструкциями.

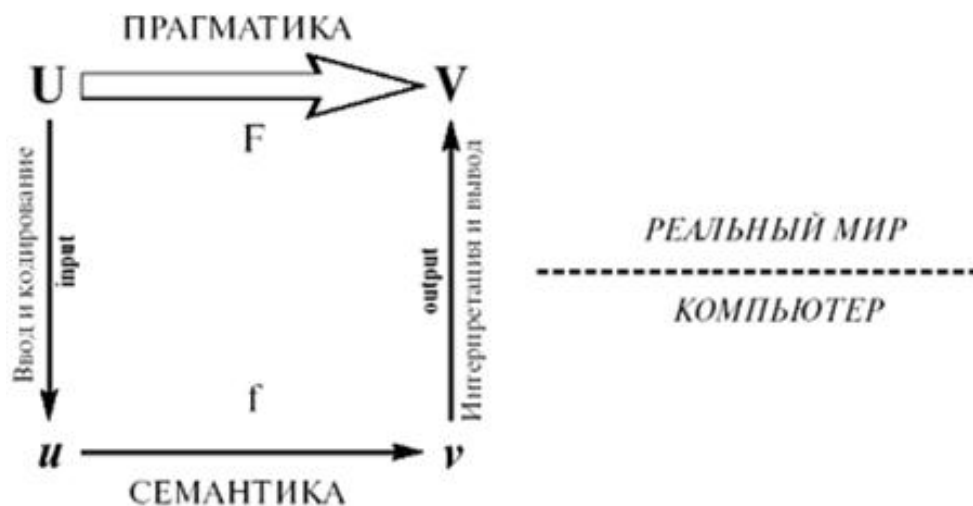


Рисунок 9.18 – Семантика (смысл программы с точки зрения выполняющего ее компьютера) и прагматика (смысл программы с точки зрения ее пользователей)

Модель содержит не все признаки и свойства представляемого ею предмета или понятия, а только те, которые существенны для разрабатываемой программной системы. Таким образом, модель беднее, а следовательно, проще представляемого ею предмета или понятия.

Простота модели по отношению к реальному предмету позволяет сделать ее формальной. Благодаря такому характеру моделей при разработке можно четко выделить все зависимости и операции над ними в создаваемой программной системе. Это упрощает как разработку и изучение (анализ) моделей, так и их реализацию на компьютере.

Объектно-ориентированный подход обладает такими *преимуществами*:

- уменьшение сложности и повышение надежности программного обеспечения;
- возможность модификации отдельных компонентов программного обеспечения без изменения остальных;

– повторное использование отдельных компонентов программного обеспечения.

Систематическое применение объектно-ориентированного подхода позволяет разрабатывать хорошо структурированные, надежные в эксплуатации, достаточно просто модифицируемые программные системы. Этим объясняется интерес программистов к объектно-ориентированному подходу и объектно-ориентированным языкам программирования. ООП является одним из наиболее интенсивно развивающихся направлений теоретического и прикладного программирования.

### **Объекты**

Объект – это понятие, абстракция или любой предмет с четко очерченными границами, имеющий смысл в контексте рассматриваемой прикладной проблемы. Введение объектов преследует две цели:

- понимание прикладной задачи (проблемы);
- введение основы для реализации на компьютере.

Каждый объект имеет определенное время жизни. В процессе выполнения программы или функционирования какой-либо реальной системы могут создаваться новые объекты и уничтожаться уже существующие.

Каждый объект имеет состояние, обладает четко определенным поведением и уникальной идентичностью.

**Состояние** (*state*) – совокупный результат поведения объекта: одно из стабильных условий, охарактеризованных количественно, в которых объект может существовать. В любой момент времени состояние объекта включает перечень (обычно статический) свойств объекта и текущие значения (обычно динамические) этих свойств.

**Поведение** (*behavior*) – действия и реакции объекта, выраженные в терминах передачи сообщений и изменения состояния; видимая извне и воспроизводимая активность объекта.

Для каждого объекта существует определенный набор действий, которые с ним можно произвести. Результат выполнения действий зависит от состояния объекта на момент совершения действия.

Программа, написанная с использованием ООП, обычно состоит из множества объектов, и все эти объекты взаимодействуют между собой. Обычно говорят, что взаимодействие между объектами в программе происходит посредством передачи сообщений между ними.

Программу, построенную по технологии ООП, можно представить себе как виртуальное пространство, заполненное объектами, которые условно «живут» некоторой жизнью. Их активность проявляется в том, что они вызывают друг у друга методы, или посылают друг другу сообщения. Внешний интерфейс объекта, или набор его методов, – это описание того, какие сообщения он может принимать.

**Уникальность** (*identity*) – свойство объекта; то, что отличает его от других объектов.

Например, у вас может быть несколько одинаковых монет. Даже если абсолютно все их свойства (атрибуты) одинаковы (год выпуска, номинал и т. д.) и при этом вы можете использовать их независимо друг от друга, они по-прежнему остаются разными монетами.

В машинном представлении под параметром уникальности объекта чаще всего понимается адрес размещения объекта в памяти. На один объект может указывать несколько ссылок, и ссылки могут менять свои значения (ссылаться на другие объекты).

**Класс** – это шаблон поведения объектов определенного типа с заданными параметрами, определяющими состояние. Все экземпляры одного класса (объекты, порожденные от одного класса) имеют один и тот же набор свойств и общее поведение, то есть одинаково реагируют на одинаковые сообщения.

Все объекты одного и того же класса описываются одинаковыми наборами атрибутов. Однако объединение объектов в классы определяется не наборами атрибутов, а семантикой. В классе вводятся имена атрибутов, которые определены для объектов. В этом смысле описание класса аналогично описанию типа структуры или записи (*record*), широко применяющихся в процедурном программировании; при этом каждый объект имеет тот же смысл, что и экземпляр структуры (переменная или константа соответствующего типа).

В соответствии с UML (*Unified Modelling Language* – унифицированный язык моделирования) класс имеет следующее графическое представление (рис. 9.19).

Класс изображается в виде прямоугольника, состоящего из трех частей. В верхней части помещается название класса, в средней – свойства объектов класса, в нижней – действия, которые можно выполнять с объектами данного класса (методы).

Название класса
Свойства объектов класса
Выполняемые действия (методы) с объектами данного класса

Рисунок 9.19 – Графическое представление класса

Каждый класс также может иметь **специальные методы**, которые автоматически вызываются при создании и уничтожении объектов этого класса:

- конструктор (*constructor*) – выполняется при создании объектов.
- деструктор (*destructor*) – выполняется при уничтожении объектов.

Обычно конструктор и деструктор имеют специальный синтаксис, который может отличаться от синтаксиса, используемого для написания обычных методов класса.

**Инкапсуляция** (*encapsulation*) – это сокрытие реализации класса и отделение его внутреннего представления от внешнего (интерфейса).

При использовании объектно-ориентированного подхода не принято применять прямой доступ к свойствам какого-либо класса из методов других классов. Для доступа к свойствам класса принято задействовать специальные методы этого класса для получения и изменения его свойств.

Внутри объекта данные и методы могут обладать различной **степенью открытости** (или доступности).

Открытые члены класса составляют внешний интерфейс объекта. Это та функциональность, которая доступна другим классам. Закрытыми обычно объявляются все свойства класса, а также вспомогательные методы, которые являются деталями реализации и от которых не должны зависеть другие части системы.

Благодаря сокрытию реализации за внешним интерфейсом класса можно менять внутреннюю логику отдельного класса, не меняя код остальных компонентов системы. Это свойство называется **модульность**.

Обеспечение доступа к свойствам класса только через его методы также дает ряд **преимуществ**.

Во-первых, так гораздо проще контролировать корректные значения полей, ведь прямое обращение к свойствам отслеживать невозможно, а значит, им могут присвоить некорректные значения.

Во-вторых, не составит труда изменить способ хранения данных. Если информация станет храниться не в памяти, а в долговременном хранилище, таком как файловая система или база данных, потребуется изменить, лишь ряд методов одного класса, а не вводить эту функциональность во все части системы.

Наконец, программный код, написанный с использованием данного принципа, легче отлаживать. Для того чтобы узнать, кто и когда изменил свойство интересующего нас объекта, достаточно добавить вывод отладочной информации в тот метод объекта, посредством которого осуществляется доступ к свойству этого объекта.

**Наследование** (*inheritance*) – это отношение между классами, при котором класс использует структуру или поведение другого класса (одиночное наследование), или других (множественное наследование) классов. Наследование вводит иерархию «общее – частное», в которой подкласс наследует от одного или нескольких более общих суперклассов.

Подклассы обычно дополняют или переопределяют унаследованную структуру и поведение.

Например, классы «Легковой автомобиль» и «Грузовой автомобиль» имеют общую функциональность: 4 колеса, двигатель, могут перемещаться и так далее. Разумно вынести эти общие свойства и функциональность в отдельный класс, например, «Автомобиль» и наследовать от него классы «Легковой автомобиль» и «Грузовой автомобиль», чтобы избежать повторного написания одного и того же кода в разных классах (рис. 9.20).

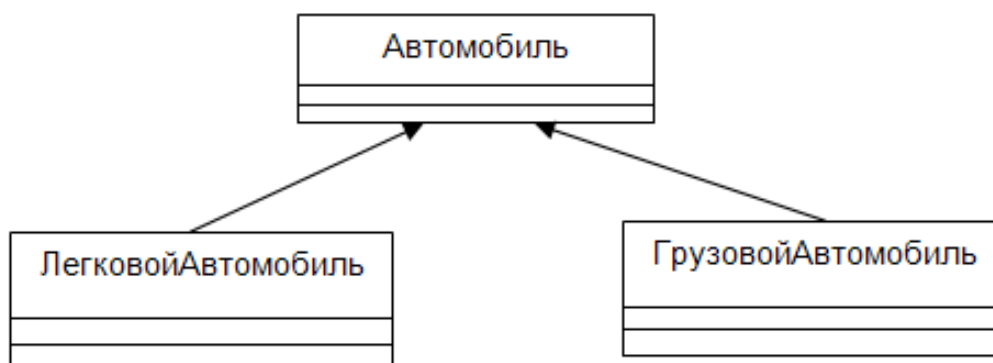


Рисунок 9.20 – Обобщение классов «Легковой автомобиль» и «Грузовой автомобиль»

Отношение обобщения обозначается сплошной линией с треугольной стрелкой на конце. Стрелка указывает на более общий класс (класс-предок или суперкласс), а ее отсутствие – на более специальный (класс-потомок или подкласс).

Использование наследования способствует уменьшению количества кода, созданного для описания схожих сущностей, а также способствует написанию более эффективного и гибкого кода.

Множественное наследование на диаграмме изображается точно так же, как одиночное, за исключением того, что линии наследования соединяют класс-потомок сразу с несколькими суперклассами.

Не все объектно-ориентированные языки программирования содержат языковые конструкции для описания множественного наследования.

**Полиморфизм** (*polymorphism*) – положение теории типов, согласно которому имена (например, переменных) могут обозначать объекты разных (но имеющих общего родителя) классов. Следовательно, любой объект, обозначаемый полиморфным именем, может по-своему реагировать на некий общий набор операций.

### **Типы отношений между классами**

Как правило, любая программа, написанная на объектно-ориентированном языке, представляет некоторый набор связанных между собой классов. Подобно тому, как стена складывается из кирпичей, компьютерная программа с использованием ООП строится из классов. Причем эти классы должны иметь представление друг о друге, для того чтобы сообща выполнять поставленную задачу.

Возможны следующие *связи между классами* в рамках объектной модели (приводятся лишь наиболее простые и часто используемые виды связей):

- агрегация (*aggregation*);
- ассоциация (*association*);
- наследование (*inheritance*);
- метаклассы (*metaclass*).

### **Агрегация**

Отношение между классами типа «содержит» (*contain*) или «состоит из» называется агрегацией (*aggregation*), или включением. Например, если компьютерная аудитория заполнена студентами, то можно сказать, что аудитория агрегирует в себе компьютеры и студентов.

Такое отношение включения, или агрегации, изображается линией с ромбиком (рис. 9.21). Необязательное название отношения записывается посередине линии



Рисунок 9.21 – Изображение агрегации

### **Ассоциация**

Если объекты одного класса ссылаются на один или более объектов другого, но ни в ту, ни в другую сторону отношение между объектами не носит характера «владения», или контейнеризации, такое отношение называют ассоциацией (*association*). Отношение ассоциации изображается так же, как и отношение агрегации, но линия, связывающая классы, – простая, без ромбика.

### **Наследование**

Наследование является важным случаем отношений между двумя или более классами. Подробно оно рассматривалось выше.

### **Метаклассы**

Итак, любой объект имеет структуру, состоящую из полей и методов. Объекты, имеющие одинаковую структуру и семантику, описываются одним классом, который и является, по сути, определением структуры объектов, порожденных от него.

В свою очередь, каждый класс, или описание, всегда имеет строгий шаблон, задаваемый языком программирования или выбранной объектной моделью. Он определяет, например, допустимо ли множественное наследование, какие существуют ограничения на именование классов, как описывают поля и методы, набор существующих типов данных и многое другое.

Таким образом, класс можно рассматривать как объект, у которого есть свойства (имя, список полей и их типы, список методов, список аргументов для каждого метода и т. д.). Также класс может обладать поведением, т. е. поддерживать методы. А раз для любого объекта существует шаблон, описывающий свойства и поведение этого объекта, значит, его можно определить и для класса. Такой шаблон, задающий различные классы, называется метаклассом.

Чтобы представить себе, что такое метакласс, рассмотрим пример некой бюрократической организации. Будем считать, что все



классы в такой системе представляют собой строгие инструкции, которые описывают, что нужно сделать, чтобы породить новый объект (например, нанять нового служащего или открыть новый отдел). Как и полагается классам, они описывают все свойства новых объектов (например, зарплату и профессиональный уровень для сотрудников, площадь и имущество для отделов) и их поведение (обязанности служащих и функции подразделений).

В свою очередь, написание новой инструкции можно строго регламентировать. Скажем, необходимо использовать специальный бланк, придерживаться правил оформления и заполнить все обязательные поля (например, номер инструкции и фамилии ответственных работников). Такая «инструкция инструкций» и будет представлять собой метакласс в ООП.

Итак, объекты порождаются от классов, а классы – от метакласса. Он, как правило, в системе только один.

### ***Достоинства объектно-ориентированного программирования***

Чтобы справиться со сложностью проектирования программ, необходимо абстрагироваться от деталей. В этом смысле классы представляют собой весьма удобный инструмент.

Классы позволяют проводить конструирование из полезных компонентов, обладающих простыми инструментами, что позволяет абстрагироваться от деталей реализации.

Данные и операции над ними образуют определенную сущность, и они не разносятся по всей программе, как нередко бывает в случае процедурного программирования, а описываются вместе. Локализация кода и данных улучшает наглядность и удобство сопровождения программного обеспечения.

Инкапсуляция позволяет привнести свойство модульности, что облегчает распараллеливание выполнения задачи между несколькими исполнителями и обновление версий отдельных компонентов.

ООП дает возможность создавать расширяемые системы. Это одно из основных достоинств ООП, и именно оно отличает данный подход от традиционных методов программирования.

Расширяемость означает, что существующую систему можно заставить работать с новыми компонентами, причем без внесения в нее каких-либо изменений. Компоненты могут быть добавлены на этапе исполнения программы.

Полиморфизм оказывается полезным преимущественно в следующих ситуациях.

Обработка разнородных структур данных. Программы могут работать, не различая вида объектов, что существенно упрощает код. Новые виды могут быть добавлены в любой момент.

Изменение поведения во время исполнения. На этапе исполнения один объект может быть заменен другим, что позволяет легко, без изменения кода, адаптировать алгоритм в зависимости от того, какой используется объект.

Реализация работы с наследниками. Алгоритмы можно обобщить настолько, что они уже смогут работать более чем с одним видом объектов.

Создание каркаса (*framework*). Независимые от приложения части предметной области могут быть реализованы в виде набора универсальных классов, или каркаса, и в дальнейшем расширены за счет добавления частей, специфичных для конкретного приложения.

Часто многоразового использования программного обеспечения не удается добиться из-за того, что существующие компоненты уже не отвечают новым требованиям. ООП помогает этого достичь без нарушения работы уже имеющихся клиентов, что позволяет извлечь максимум из многоразового использования компонентов.

Сокращается время на разработку, которое может быть отдано другим задачам.

Компоненты многоразового использования обычно содержат гораздо меньше ошибок, чем вновь разработанные, ведь они уже не раз подвергались проверке.

Когда некий компонент используется сразу несколькими клиентами, улучшения, вносимые в его код, одновременно оказывают положительное влияние и на множество работающих с ним программ.

Если программа опирается на стандартные компоненты, ее структура и пользовательский интерфейс становятся более унифицированными, что облегчает ее понимание и упрощает использование.

### **Вопросы для повторения и самоконтроля**

1. Что такое команда присваивания?
2. Как записывается команда присваивания?
3. Объясните выполнение команды присваивания.
4. Можно ли заменять знак «присвоить» знаком равенства?
5. Поясните, как изменяется величина в ряду присвоений.

6. Что такое форма записи алгоритма?
7. Какие формы записи алгоритмов вам известны?
8. Чем объясняется разнообразие форм записи?
9. Охарактеризуйте словесно-пошаговую (текстовую) форму.
10. Охарактеризуйте табличную форму записи алгоритма.
11. Как осуществляется запись на алгоритмическом языке?
12. Что такое результат выполнения алгоритма?
13. Что такое исходные данные?
14. Что представляет собой графическая запись алгоритма?
15. Охарактеризуйте основные блоки.
16. Каков порядок составления блок-схемы алгоритма?
17. Как определяются исходные данные?
18. Как определяются результаты алгоритма ?
19. Какие виды алгоритмов выделяют?
20. Охарактеризуйте линейный вид алгоритма.
21. Охарактеризуйте разветвляющийся вид алгоритма.
22. Охарактеризуйте циклический вид алгоритма.
23. Какой алгоритм называют комбинированным?
24. Какие структуры может включать в себя алгоритм?
25. Охарактеризуйте данные структуры.
26. Какими ключевыми словами можно пользоваться при определении вида алгоритма?
27. Для чего необходимо ветвление в алгоритмах?
28. Что такое условие?
29. Какие формы ветвления различают?
30. Сравните формы ветвления между собой.
31. Как оформляют ветвление в алгоритмах, записанных в виде блок-схемы?
32. Как оформляют ветвление в алгоритмах, записанных на алгоритмическом языке?
33. Как осуществляют выполнение действий в ветвлении при записи алгоритма на алгоритмическом языке?
34. Какими формулами пользуются при нахождении суммы или произведения ряда чисел? Охарактеризуйте их.
35. Как математически записывают данные формулы?
36. Как определить по ним величину-счетчик, ее начальное и конечное значение, шаг и аргумент?
37. В чем отличие языков программирования от остальных способов составления алгоритмов?

38. Какие уровни языков программирования вам известны?
39. Деление языков высокого уровня.
40. Сущность, достоинства и недостатка процедурно-ориентированного программирования.
41. Что такое объектно-ориентированное программирование?
42. В чем заключается преимущества объектно-ориентированного подхода перед процедурно-ориентированным?
43. Что такое «объект» в объектно-ориентированном программировании?
44. Что такое «состояние» в объектно-ориентированном программировании?
45. Что такое «поведение» в объектно-ориентированном программировании?
46. Что такое «уникальность» в объектно-ориентированном программировании?
47. Что такое «класс» в объектно-ориентированном программировании?
48. Что такое «инкапсуляция» в объектно-ориентированном программировании?
49. Что такое «наследование» в объектно-ориентированном программировании?
50. Что такое «полиморфизм» в объектно-ориентированном программировании?
51. Что такое «агрегация» в объектно-ориентированном программировании?
52. Что такое «ассоциация» в объектно-ориентированном программировании?
53. Что такое «метаклассы» в объектно-ориентированном программировании?
54. Преимущества и недостатки объектно-ориентированного программирования?

### **Задания для самостоятельной работы**

1. Отметьте правильные команды присваивания знаком «+», а записанные неверно – знаком «-».

1. A:=15.
2. «13»:=x\$.
3. i:=300:2+5.
4. «год»:=M.

5.  $7.56+5:= \text{NOMER}$ .

6.  $B\$ = \langle \text{«пароход»}$ .

7.  $N:=3b+5$ .

8.  $s3 : 2*a+5*b$ .

9.  $N\$:= \text{утро}$ .

10.  $Z5:=2,7+a$ .

2. Определите, чему будут равны величины после выполнения ряда присваиваний:

1.  $A:=15; A:=A*2; A:=A-A; A:=A+20; A:=A*5$ .

2.  $B:=1; B:=B+B; B:=B*B; B:=B+B; B:=B*B$ .

3.  $N:=20; N:=N+N/2; N:=N+N; N:=N/3+N/10$ .

4.  $m:=40; m:=m-10; m:=m/2+m*2; m:=100-m$ .

5.  $a:=5; B:=7; B:=A+B; A:=A+B; B:=20+A-B$ .

3. Для приведенных ниже алгоритмов определите формы записи.

а) при  $m=7; p=2$

1. Спросить, чему равно число  $m$ .

2. Спросить, чему равно число  $p$ .

3. Вычесть из  $m$  1, результат присвоить  $d$ .

4. Разделить  $d$  на  $p$ , результат присвоить  $A$ .

5. Умножить  $p$  на  $d$ , результат присвоить  $b$ .

6. Сложить  $A$  и  $b$ , результат присвоить  $W$ .

7. Сообщить результат  $W$ .

б) при  $x=4$

Номер действия	Действие	Величина		Результат
		1	2	
1	$\wedge$	$x$	2	$a$
2	$+$	$a$	1	$b$
3	$-$	$x$	3	$c$
4	$/$	$b$	$c$	$d$
5	$/$	10	$c$	$e$
6	$-$	$d$	$e$	$y$

в) при  $p=5; q=3$

алг ВЗВ

арг  $p, q$

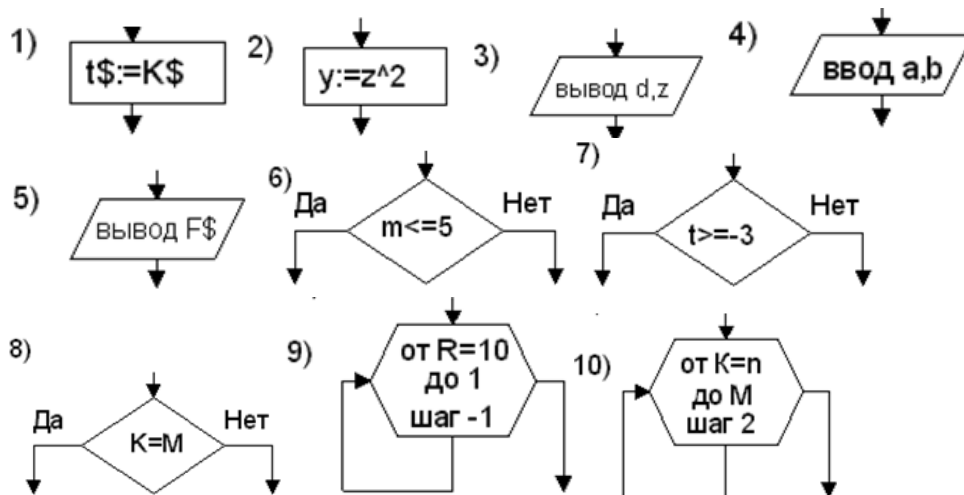
рез  $y$

нач

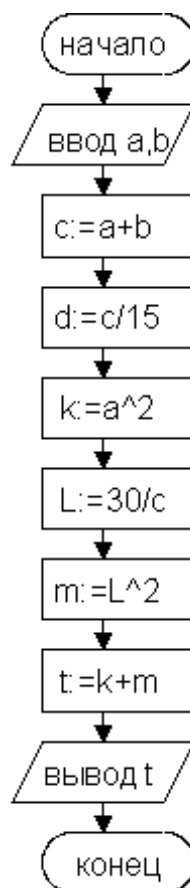
$y := (5*p) / (2*q - p) + (10*q) / (3*p - 4*q)$

кон

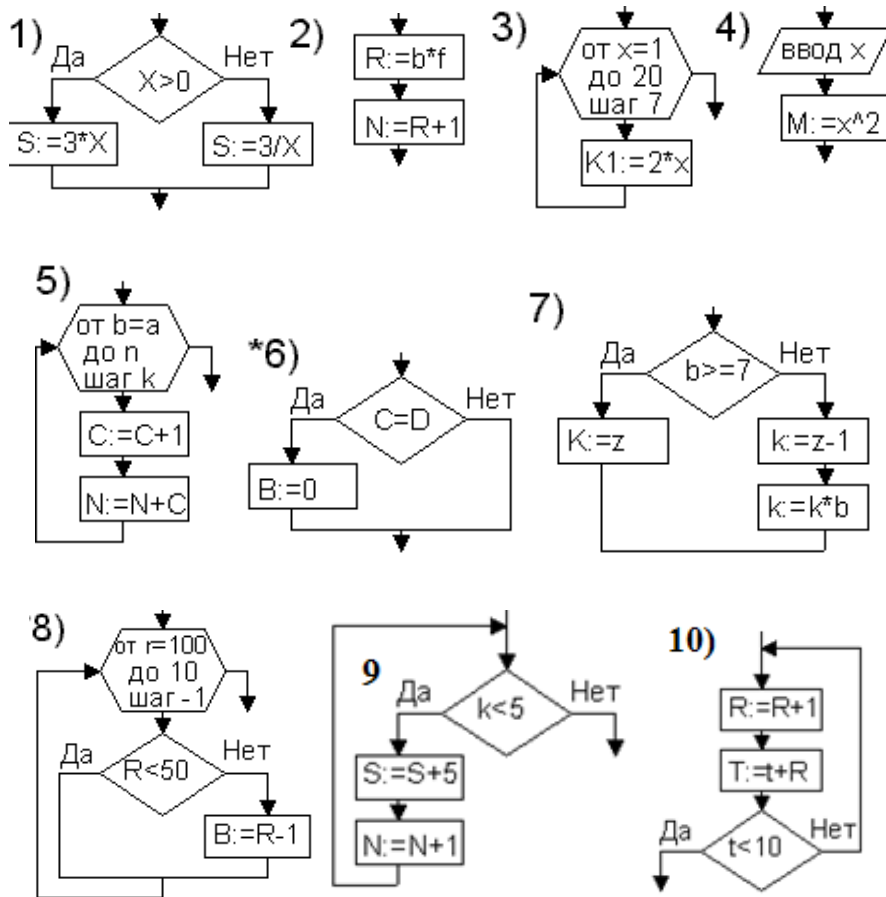
4. Восстановите по приведенным блокам действия, которые они выполняют.



5. По блок-схемам алгоритмов определите результат их выполнения при  $a=5$  и  $b=10$ . Запишите данные алгоритмы в виде таблицы, на алгоритмическом языке и в словесно-пошаговой форме.



6. По приведенным блокам определите структуры, к которым они относятся (следование, ветвление, цикл, смешанный).

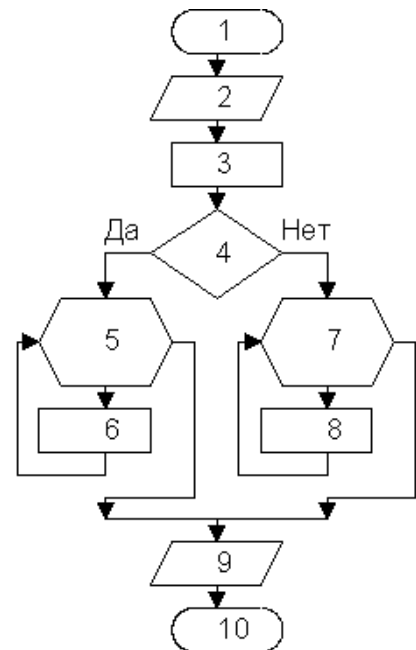


7. Расставьте в комбинированном алгоритме команды, учитывая порядок введения промежуточных результатов:

- а) вывод  $S$
- б) от  $R=n$  до  $m$  шаг 1
- в) начало
- г)  $S := S + 1$
- д)  $n < m$
- е)  $S := S * T$
- ж)  $S := 1$
- з) от  $T=n$  до  $m$  шаг -1
- и) конец
- к) ввод  $n, m$

8. Опишите данный комбинированный алгоритм по блокам, входящим в состав его структур:

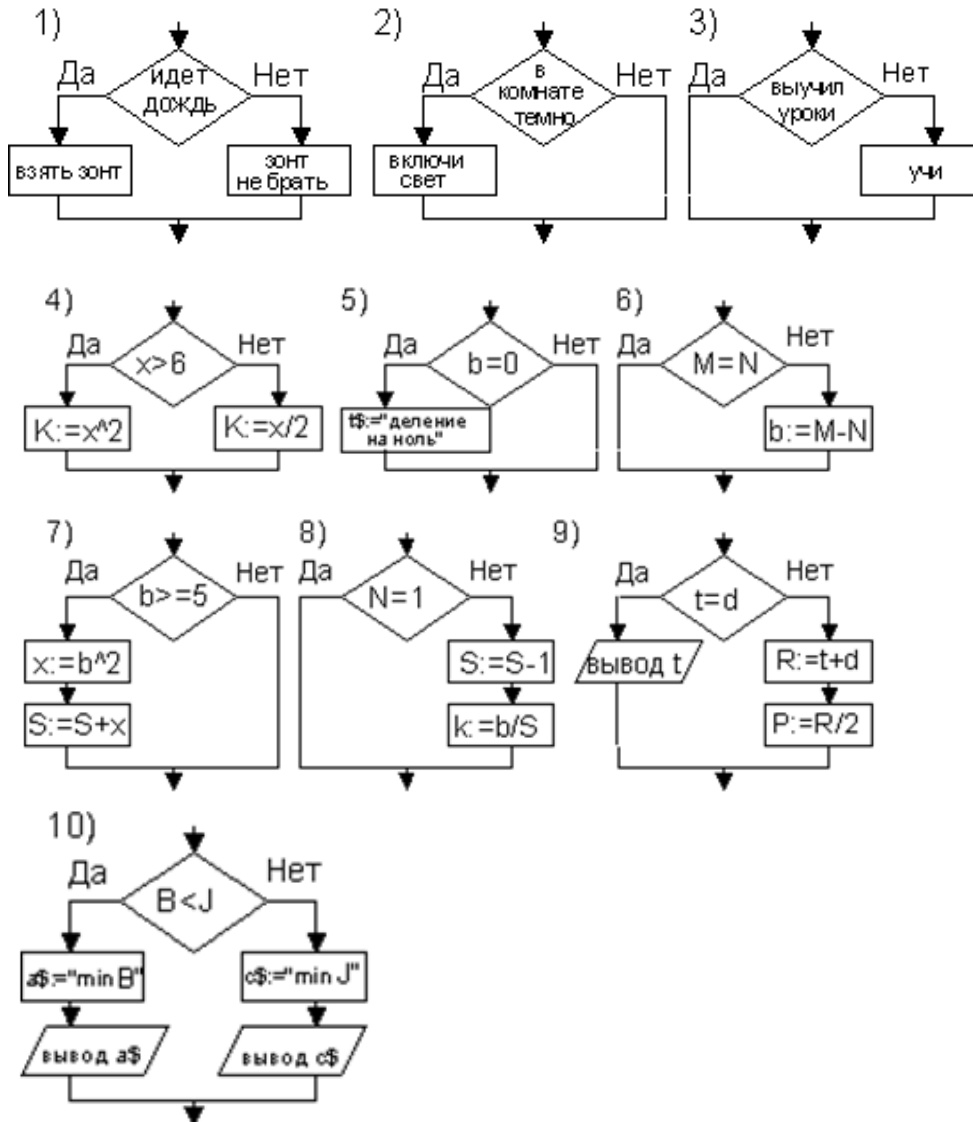
- а) следование:
- б) ветвление:
- в) цикл:



9. В банк был положен вклад в размере N рублей. Через три года он увеличился на 50 %. Сколько денег получил вкладчик? Составьте алгоритм решения задачи.

10. Сберегательный банк 20 месяцев хранил вклад под 2 % в месяц и 16 месяцев под 5 % в месяц. Определите размер вклада на момент его получения. Составьте алгоритм решения задачи.

11. Для приведенных блок-схем задайте исходные условия и опишите, как будут выполняться данные алгоритмы. Запишите эти структуры в формульно-словесном виде или алгоритмическом языке.



12. Создайте алгоритм решения в блок-схемном и формульно-словесном виде для следующих заданий:

$$1) M = \begin{cases} \frac{5}{6}c + \frac{3}{8}d - 3 & \text{при } c > 1 \\ \frac{7}{8}d - \frac{3}{2}c + 2 & \text{при } c \leq 1 \end{cases}$$

$$2) S = \begin{cases} -1 & \text{при } x < 0 \\ 0 & \text{при } x = 0 \\ 1 & \text{при } x > 0 \end{cases}$$



$$3) z = \begin{cases} x^2 + y^2 & , \text{ если } x^2 + y^2 \leq 1, \\ x + y & , \text{ если } x^2 + y^2 > 1 \text{ и } y \geq x, \\ 0,5 & , \text{ если } x^2 + y^2 > 1 \text{ и } y < x; \end{cases}$$

$$4) v = \begin{cases} x + y & , \text{ если } x > 1 \text{ и } y > 1, \\ x - y & , \text{ если } x > 1 \text{ и } y \leq 1, \\ -x + y & , \text{ если } x \leq 1 \text{ и } y > 0, \\ -x - y & , \text{ если } x \leq 1 \text{ и } y \leq 0. \end{cases}$$

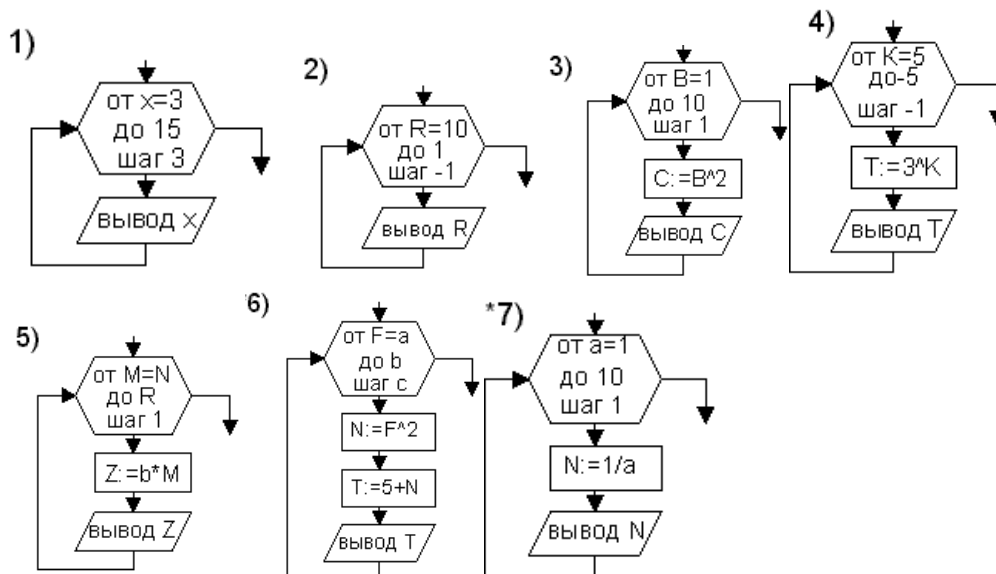
$$5) z = \begin{cases} 1 & , \text{ если } c = 0, \\ x & , \text{ если } c = 1, \\ 3x^2 - 1/2 & , \text{ если } c = 2, \\ x^3 - 3x/2 & , \text{ если } c = 3, \\ 2x^4 - 3x/2 & , \text{ в противном случае} \end{cases}$$

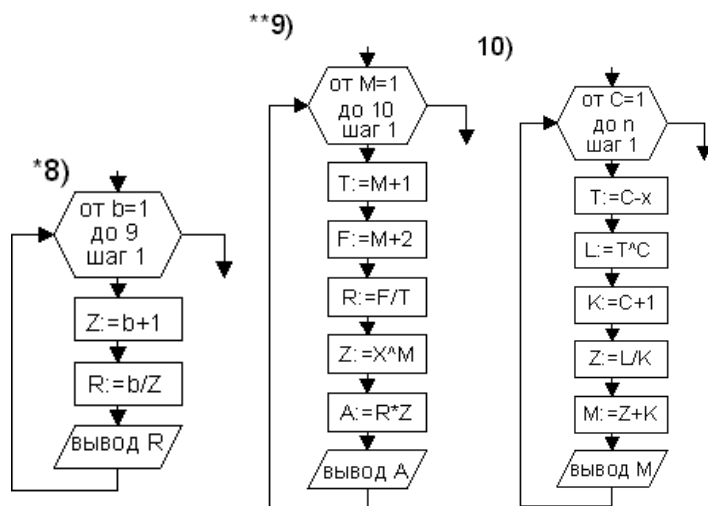
13. Определите, пройдет ли колобок с длиной талии  $L$  в квадратное окно со стороной  $A$ .

14. В кафе продают лимонад по  $N$  руб. и мороженое по  $T$  руб. Составьте алгоритм, который сообщает «возьмите сдачу», «доплатите еще» или «спасибо за покупку» при оплате покупателем за  $M$  лимонадов и  $R$  мороженых денег в размере  $S$  руб.

15. Успеет ли спортсмен добраться до финиша, расположенного за  $S$  км, если он  $a$  часов едет на велосипеде со скоростью  $V1$  км/ч и  $b$  часов бежит со скоростью  $V2$  км/ч.

16. По блок-схеме цикла с параметром определите, какие будут выведены результаты (математический ряд).





17. Составьте блок-схемы циклов, записанных на алгоритмическом языке. Какие значения принимает величина-счетчик в данных циклах? Какие результаты будут выведены (в математической записи ряда)?

1)

для  $t=5$  до 15 шаг 5

нц

$R:=t - 5$

вывод  $R$

кц

3)

для  $F=5$  до 10 шаг 1

нц

$S:=(F+1)/(F-1)$

вывод  $S$

кц

5)

для  $M=12$  до 3 шаг -3

нц

$F:=M*(M+5)/(M-5)$

вывод  $F$

кц

2)

для  $m=10$  до -10 шаг -2

нц

$t:=m+1$

вывод  $t$

кц

4)

для  $a=2$  до -2 шаг -1

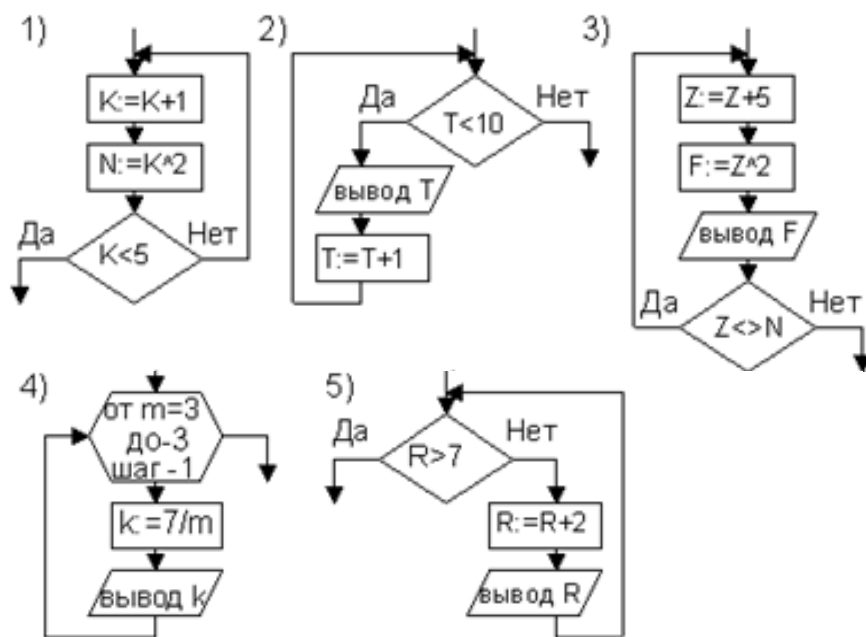
нц

$t:=a^a+a$

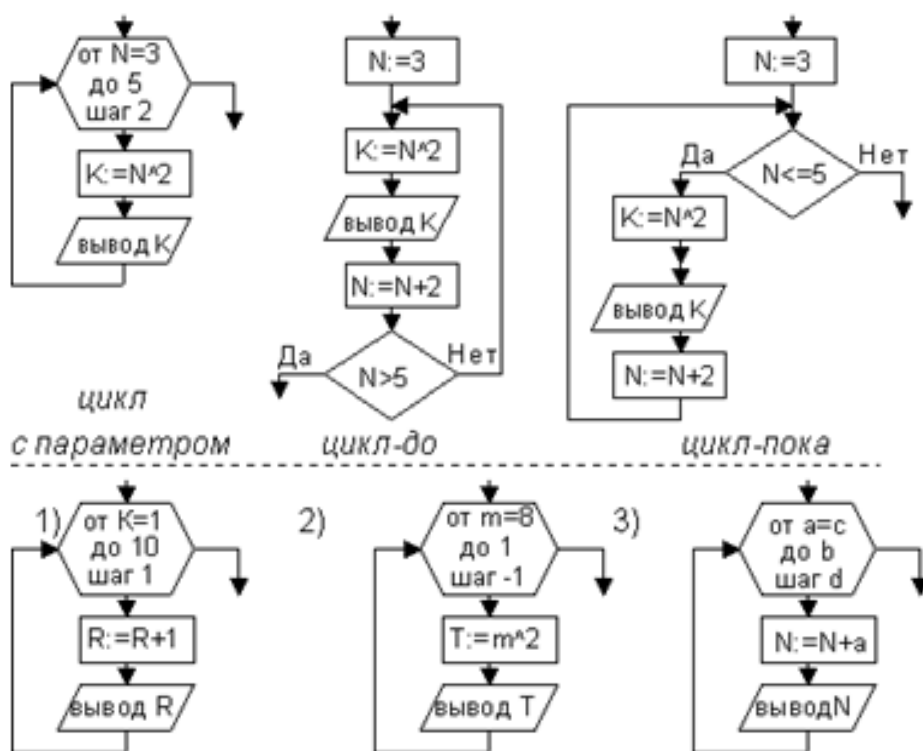
вывод  $t$

кц

18. Определите по блок-схемам нижеприведенных структур, к какому виду циклов они относятся? Запишите данные структуры на алгоритмическом языке.



19. Преобразуйте циклы с параметром в «цикл – до» и «цикл – пока». Запишите их в виде блок-схем. Образец:



20. В банк был положен вклад в размере  $N$  руб. Каждый месяц он увеличивался на 1%. Определите размер вклада через 5 лет. Запишите алгоритм задачи в блок-схемном виде.

21. Составьте алгоритмы в виде блок-схем.

$$1) S = \sum_{i=2}^{10} k \quad 2) P = \prod_{i=2}^{10} b+i \quad 3) T = \prod_{i=2}^{10} \frac{k+2}{k}$$

$$4) R = \sum_{m=n}^f \frac{1}{c_m} \quad 5) \frac{1}{S+2} + \frac{1}{S+3} + \dots + \frac{1}{S+h}$$

$$6) a \cdot 3a \cdot 5a \cdot \dots \cdot 11a.$$

22. Определите количество четных чисел, начиная от 1 до 157. Сколько будет таких чисел? Каких чисел будет больше: четных или нечетных? Составьте алгоритмы в виде блок-схем.

23. Найдите произведение всех четных чисел от 50 до 500. Составьте алгоритмы в виде блок-схем.

24. В группе N учеников. Их рост заносится в ЭВМ. Рост девушек кодируется положительным числом, а юношей – отрицательным. Определите средний рост девушек. Составьте алгоритмы в виде блок-схем.

25. По условию предыдущей задачи определите средний рост юношей. Составьте алгоритмы в виде блок-схем.

26. Выдайте сообщение о том, кто выше в данном классе (сравнивая средний рост девушек и юношей). Составьте алгоритмы в виде блок-схем.

27. Определите значение целочисленной переменной S после выполнения операторов:

S:=128 нц для i от 1 до 4 S:=div(S,2) кц		1) S:=1; a:=1 нц для i от 1 до 3 S:=S+i*(i+1)*a a:=a+2 кц <b>Ответ</b>
Решение:		
i	S	2) S:=1; a:=1 нц для i от 1 до 3 S := S+i нц для j от 2 до 3 S := S+j кц кц <b>Ответ</b>
1	128	
2	128/2=64	
3	64/2=32	
4	32/2=16	
	16/2=8	
Ответ: S=8		

S:=0 нц для i от 1 до 2 нц для j от 2 до 3 S:=S+i+j кц кц			3) нц для i от 1 до 3 S:=0 нц для j от 2 до 3 S:=S+i+j кц кц <b>Ответ</b>
Решение			
i	j	S	4) нц для i от 1 до 2 S := 0 нц для j от 2 до 3 нц для k от 1 до 2 S := S+i+j+k кц кц кц <b>Ответ</b>
		0	
1	2	0+1+2=3	
	3	3+1+3=7	
2	2	7+2+2=11	
	3	11+2+3=16	
Ответ: S=16			

28. Определите значение переменной S после выполнения операторов:

i:=0; S:=0	S:=0; N:=125
нц пока i<3	нц пока N>0
i:=i+1;	S:=S+mod(N,10)   S – сумма цифр
S:=S+i*i	N:=div(N,10)   числа N
кц	кц

Решение			Решение		
Условие $i < 3$	i	S	Условие $N > 0$	S	N
	0	0		0	125
$0 < 3?$ да	1	$0+1^2=1$	$125 > 0?$ да	$0+5=5$	12
$1 < 3?$ да	2	$1+2^2=5$	$12 > 0?$ да	$5+2=7$	1
$2 < 3?$ да	3	$5+3^2=14$	$1 > 0?$ да	$7+1=8$	0
$3 < 3?$ нет(кц)			$0 > 0?$ нет (кц)		
Ответ: S=14			Ответ: S=8		

1) S:=0; i:=1  
нц пока i>1  
S:=S+1/i  
i:=i-1  
кц

2) S:=0; i:=1; j:=5  
нц пока i<j  
S:=S+i\*j  
i:=i+1  
j:=j-1  
кц

3) a:=1; b:=1; S:=0;  
нц пока a<=5  
a:=a+b; b:=b+a;  
S:=S+a+b  
кц

4) a:=1; b:=1  
нц пока a+b<10  
a:=a+1  
b:=b+a  
кц  
S:=a+b

*Ответы к заданиям для самостоятельной работы*<sup>13</sup>

1. Верные ответы: правильное написание: 1; 4; 6; 7; 10  
неправильное написание: 2; 3; 5; 8; 9.

2. Верные ответы: 1. A = 100; 2. B = 64; 3. N = 26; 4. m = 25;  
5. A = 17, B = 25.

3. Верные ответы: а) текстовая форма; б) табличная форма;  
в) алгоритмический язык.

<sup>13</sup>На ряд заданий ответы отсутствуют, решение надо найти самостоятельно.

4. Верные ответы: 1,2 – присваивание; 3, 4, 5 – ввод – вывод; 6, 7, 8 – ветвление; 9, 10 – цикл.
5. Верные ответы:  $t:=29$ .
6. Верные ответы: 2, 4 – следование; 1, 6, 7 – ветвление; 3, 5, 8 – цикл; 9, 10 – комбинированный.
7. Верные ответы: 1 – в; 2 – к; 3 – ж; 4 – д; 5 – б; 6 – г; 7 – з; 8 – е; 9 – к; 10 – и.
8. Верные ответы: 1, 2, 3, 9, 10 – следование; 4 – ветвление; 5, 6, 7, 8 – цикл.
9. Верные ответы:  $1,5 * N$
10. Верные ответы: 120 %
16. Верные ответы: 1) 3; 6;...12; 15. 2) 10; 9;...2; 1. 3) 1; 4;...81; 100. 4) 243;...0,004115. 7) 1; 0,5;...0,111; 0,1. 8) 0,5; 0,667;...0,889; 0,9.
17. Верные ответы: 1)  $R:= 0; 5; 10$ . 2)  $t:= 11; 9;...-7; -9$ . 3)  $S:= 7,991$ . 4)  $t:= 6; 2;...-2; -1,75$ . 5)  $F:= 29; 14;...66; -12$ .
18. Верные ответы: 2, 5 – цикл «до»; 1, 3 – цикл «пока»; 4 – цикл с параметрами.
20. Верные ответы:  $N_{нач} = 81,67 \%$
22. Верные ответы: Четных чисел 78 на одно меньше чем нечетных.
27. Верные ответы: 1) 0; 2) 13; 3) 52; 4) 14.

## Глава 10. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

### 10.1. Классификация программного обеспечения

Бурное развитие новой информационной технологии и расширение сферы ее применения привели к интенсивному развитию программного обеспечения (ПО).

Под программным обеспечением информационных систем понимается совокупность программных и документальных средств для создания и эксплуатации систем обработки данных средствами вычислительной техники. В компьютерном жаргоне часто используется слово «софт» (от английского *software*).

По назначению программное обеспечение разделяется на системное, прикладное и инструментальное (рис. 10.1, 10.2, 10.3).



Рисунок 10.1 – Иерархия программного обеспечения

Инструментальное ПО (среды программирования) предназначено для использования в ходе проектирования, разработки и сопровождения программ. Это интегрированные среды разработки, SDK, компиляторы, интерпретаторы, линковщики, ассемблеры, отладчики и т. п.

Прикладное ПО (приложения) – программы, предназначенные для выполнения определенных пользовательских задач и рассчитанные на непосредственное взаимодействие с пользователем. К прикладному ПО относятся банковские и бухгалтерские программы, игры, текстовые и графические редакторы, Интернет-клиенты и т. п.

Базовое (системное) ПО используется для обеспечения работы компьютера самого по себе и выполнения прикладных программ. Конкретные виды системного программного обеспечения включают

загрузчики, операционные системы, драйверы устройств, утилиты (сервисные программы). Наиболее общая часть системного программного обеспечения – операционная система.



Рисунок 10.2 – Классификация программного обеспечения

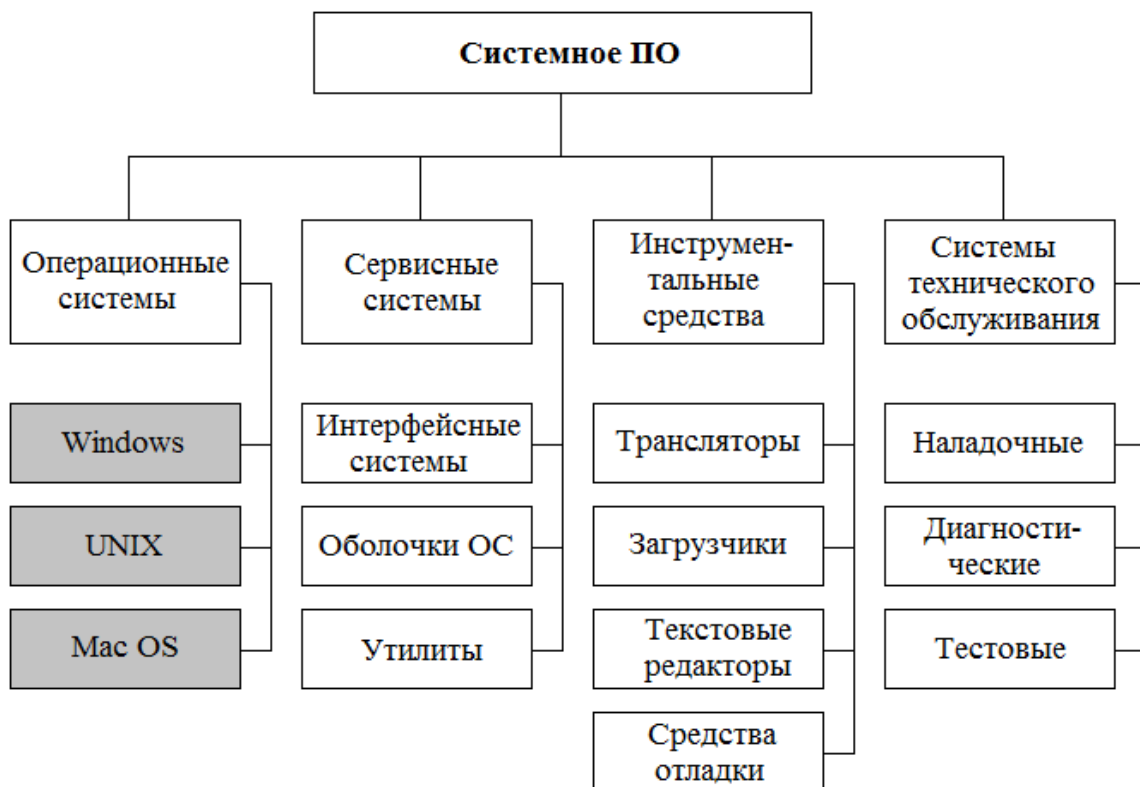


Рисунок 10.3 – Классификация системного программного обеспечения



## 10.2. Операционные системы

Операционная система – комплекс программ, постоянно (псевдопостоянно) находящихся в памяти ЭВМ, организующий управление устройствами машины и ее взаимодействие с пользователем (интерфейс).

В операционную систему обычно входят следующие программы: стартовая программа, диспетчер (монитор или супервизор) (очередность исполнения программ), редакторы, загрузчики (для ввода программ в ОЗУ), файловая система, административная система (учет ресурсов), а также базовое программное обеспечение.

Операционная система (ОС) – комплекс программных средств, который загружается при включении компьютера и обеспечивает:

- загрузку в оперативную память и выполнение всех программ;
- управление ресурсами компьютера (оперативной памятью, процессорным временем, файловой системой, внешними устройствами);
- управление оперативной памятью;
- управление энергонезависимой памятью (жесткий диск, компакт-диски и т. д.), как правило, с помощью файловой системы;
- диалог пользователя с компьютером, предоставляя удобный способ взаимодействия (интерфейс).

К дополнительным функциям ОС можно отнести:

- параллельное или псевдопараллельное выполнение задач (многозадачность);
- взаимодействие между процессами;
- межмашинное взаимодействие (компьютерная сеть);
- защита самой системы, а также пользовательских данных и программ от вредоносных действий пользователей или приложений;
- разграничение прав доступа и многопользовательский режим работы (аутентификация, авторизация).

Интерфейс – это совокупность средств и правил, которые обеспечивают взаимодействие устройств, программ и человека. В зависимости от объектов взаимодействия интерфейс определяют как пользовательский, аппаратный, программный. Например, интерфейс между пользователем и программно-аппаратными средствами компьютера называют пользовательским интерфейсом, а между аппаратным и программным обеспечением – аппаратно-программный интерфейс.

### *Открытое, закрытое и свободное программное обеспечение*

Говоря о создании и распространении программного обеспечения, следует отметить две основные стратегии, применяемые в этой области.

Собственническое, или проприетарное, программное обеспечение (*proprietary software*) – это программное обеспечение, которое имеет ограничения на его использование и копирование, требуемые собственником. Предотвращение использования, копирования или модификации могут быть достигнуты правовыми или техническими средствами.

Технические средства включают в себя выпуск только машинно-читаемых двоичных файлов и сокрытие читаемого человеком исходного кода.

Правовые средства могут включать в себя лицензирование ПО, копирайт (*copyright*, ©) и патенты.

По мнению многих, программирование – это такая же наука, как химия, физика, математика. Все достижения в этих областях обнаружены. Если человек живет в обществе, то все его открытия должны стать достоянием этого общества, ведь именно так происходит прогресс. То же можно сказать и о программном обеспечении.

В семидесятых годах XX века программное обеспечение зачастую разрабатывалось свободными объединениями программистов и бесплатно передавалось другим нуждающимся в нем пользователям. Нередко этим занимались даже крупные фирмы. К 1983 г. положение изменилось – наступила эра персональных компьютеров, коммерческие программы и операционные системы (в частности, DOS от Microsoft) начали свое победное шествие по миру.

Ричард Столлмен, один из основателей ОС Unix в 1983 г., основал проект GNU, целью которого было вернуть прежние взаимоотношения производителей и потребителей программного обеспечения. GNU (рекурсивный акроним от англ. GNU is Not Unix – «GNU – это не Unix») – проект по созданию свободной UNIX-подобной операционной системы.

В рамках проекта GNU в 1988 г. Столлмен совместно с юристами разработал лицензию GNU General Public License (Открытое лицензионное соглашение GNU) для программного обеспечения.

Получив в пользование или купив свободную программу, вы легально, не нарушая ничьих авторских прав, можете свободно:

- читать (изучать) код;
- писать (модифицировать) код;
- распространять (публиковать, тиражировать) код.

Вы не имеете права скрыть исходный код программы, объявив себя его владельцем, и остановить свободное совершенствование и развитие программы.

Программное обеспечение с *открытым* исходным кодом (*Open source*) – способ разработки ПО, при котором создаваемый исходный код программ открыт, т. е. общедоступен для просмотра и изменения. Это позволяет всем желающим использовать уже созданный код для своих нужд и помогать в разработке открытой программы.

Открытая лицензия не требует, чтобы ПО всегда предоставлялось бесплатно. Но многие из наиболее успешных проектов открытого ПО бесплатны.

Подавляющее большинство открытых программ является одновременно *свободными* (GNU GPL) и наоборот, ибо определения открытого и свободного ПО близки, а большинство лицензий соответствуют обоим.

Отличие между движениями *открытого* и *свободного* ПО заключается в основном в приоритетах. Сторонники открытого ПО делают упор на эффективность открытых исходных кодов программ как метода разработки. Сторонники свободного ПО исходят из идеологических соображений, и считают, что именно права на распространение, модификацию и изучение программ являются главным достоинством свободного ПО.

#### *Семейства операционных систем*

Наиболее древняя из успешных операционных систем – UNIX (1969 г.). Она до сих пор остается одной из основных систем на компьютерах, которые мощнее персональных и от нее порождено множество UNIX-подобных ОС.

В 1981 г. фирма IBM выпустила персональный компьютер (IBM PC), архитектура которого стала стандартом в мире. Все персональные компьютеры принято разделять на IBM-совместимые (абсолютное большинство) и IBM-несовместимые, например компьютер Macintosh, производимый фирмой Apple. Компьютеры Macintosh работают под управлением операционной системы MacOS.

#### *ОС Microsoft Windows*

В 1982 г. – фирма Microsoft выпустила ставшую на долгие годы самой популярной для IBM-совместимых ПК операционную систему MS-DOS (*Disc Operating System*).

Начиная с 1995 г., Windows – самая популярная операционная система на рынке персональных компьютеров – стандарт де-факто.

К 2005 г. Microsoft Windows была установлена более чем на 89 % персональных компьютеров.

Однако многие пользователи сделали выбор в пользу ОС Windows, так как совсем не знакомы с альтернативами, такими как MacOS, Linux, BSD.

В России до начала 2000 гг. почти все персональные компьютеры продавались с предустановленной операционной системой Windows. Борьба с распространением пиратских версий программных продуктов привела к появлению интереса к другим операционным системам. Так, стало возможным приобрести персональный компьютер с предустановленной бесплатной ОС Linux.

#### *Хронология выпусков версий Windows*

1. Windows 1.0 (1985).
2. Windows 2.0 (1987).
3. Windows/386 (1987).
4. Windows 3.0 (1990).
5. Windows 3.1 (1992).
6. Windows NT 3.1 (1993).
7. Windows NT 3.5 (1994).
8. Windows 95 (1995).
9. Windows NT 4.0 (1996).
10. Windows 98 (1998).
11. Windows 2000 (2000).
12. Windows Me (2000).
13. Windows XP (2001).
14. Windows 2003 (2003).
15. Windows Vista (2007).
16. Windows рабочее название Vienna (2010).

Операционная система Microsoft Windows включает в себя стандартные приложения: браузер Internet Explorer, почтовый клиент Outlook Express, проигрыватель Windows Media Player. Вокруг факта включения таких стандартных продуктов в ОС Windows разгорается много споров и дискуссий, поскольку это создает серьезное препятствие для распространения конкурирующих продуктов.

Для MS Windows существует очень удобный и освоенный большинством пользователей пакет прикладных программ Microsoft Office (рис.10.4), включающий:

- текстовый процессор MS Word;
- табличный процессор MS Excel;

- органайзер MS Outlook;
- приложение для подготовки презентаций MS PowerPoint;
- приложение для управления базами данных MS Access.



*Рисунок 10.4 – Логотипы прикладных программ пакета Microsoft Office*

Не стоит забывать, что и ОС Windows и абсолютное большинство популярных прикладных программ под Windows имеют лицензию *copyright*, т. е. являются проприетарным ПО. Соответственно каждая копия такой программы должна приобретаться за деньги. Например, на конец 2006 года цена на Windows XP Home Edition Russian BOX (SP2) – \$185, Office 2003 Russian CD – \$322. А новые программные продукты Microsoft на конец 2007 года уже составляли: Microsoft Vista – \$280, Microsoft Office Professional Plus 2007 – \$500.

#### *ОС GNU/Linux*

Операционная система GNU/Linux – свободная UNIX-подобная операционная система (рис. 10.5). Это самая мощная альтернатива MS Windows, популярная в качестве серверной и резко набирающая популярность в качестве настольной (*desktop*) операционной системы в последние годы, в связи с усилением контроля за соблюдением лицензионного использования ОС Windows со стороны Microsoft в версиях XP и Vista.



*Рисунок 10.5 – Логотип ОС GNU/Linux – пингвиненок Тих*

К операционной системе GNU/Linux также часто относят программы, дополняющие эту операционную систему, и прикладные программы, делающие ее полноценной многофункциональной операционной средой.

В отличие от большинства других операционных систем, GNU/Linux не имеет единой официальной комплектации. Вместо этого GNU/Linux поставляется в большом количестве так называемых дистрибутивов, в которых программы GNU соединяются с ядром Linux и другими программами. Самые распространенные в мире дистрибутивы:

- американский Red Hat и его наследник Fedora Core;
- немецкий SuSE;
- французский Mandriva (бывший Mandrake);
- не имеющий национальной принадлежности международный дистрибутив Debian GNU/Linux;
- один из самых старых дистрибутивов Slackware;
- сравнительно молодой и активно развивающийся дистрибутив Gentoo;
- молодой, но перспективный дистрибутив Ubuntu Linux.

GNU/Linux не имеет географического центра разработки. Нет и организации, которая владела бы этой системой; нет даже единого координационного центра. Программы для GNU/Linux – результат работы тысяч проектов. Некоторые из этих проектов централизованы, некоторые сосредоточены в фирмах, но большинство объединяют

программистов со всего света, которые знакомы только по переписке. Создать свой проект или присоединиться к уже существующему может любой и, в случае успеха, результаты работы станут известны миллионам пользователей. Пользователи принимают участие в тестировании свободных программ, общаются с разработчиками напрямую, что позволяет быстро находить и исправлять ошибки и реализовывать новые возможности.

На рынке серверных операционных систем Австралии Linux уже заняла около 30 %. За австралийцами последовали бразильцы. Бразильское правительство решило отказаться от расходования средств на программное обеспечение и перейти с продуктов Microsoft на системы с открытым кодом, в частности, ОС Linux. Главная причина изменений – экономическая. В нашей стране рынок Linux пока невелик.

### 10.3. Файлы и файловая система

Вся информация в компьютере хранится в файлах, с которыми и работает операционная система.

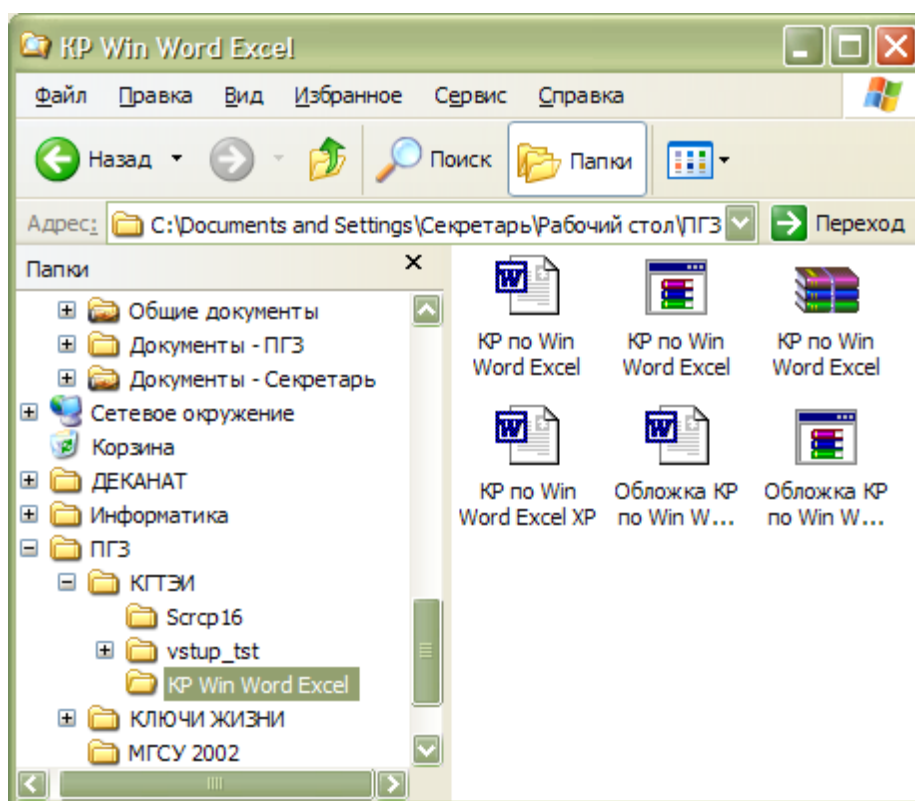
Файл (*file*) – это поименованная последовательность байтов, место постоянного хранения различных видов информации.

Файлы организованы в каталоги, также называемые директориями (*directory*) или папками (*folder*). Каталог – это файл, который хранит сведения о других файлах: имя, размер, дату создания, адрес файла на диске.

Любой каталог может содержать произвольное число подкаталогов, в каждом могут храниться файлы и другие каталоги.

На каждом диске существует главный или корневой каталог, в котором располагаются все остальные каталоги, называемые подкаталогами и некоторые файлы. Таким образом, создается иерархическая структура (рис. 10.6). Каталог, с которым в настоящий момент работает пользователь, называется текущим.

Файлы и каталоги являются самыми важными объектами файловой системы. Файловая система – регламент, определяющий способ организации, хранения и именования данных на носителях. Она определяет формат физического хранения файлов. Конкретная файловая система определяет размер имени файла, максимально возможный размер файла, набор атрибутов файла.



*Рисунок 10.6 – Иерархическая структура файлов*

В каждом каталоге (разветвлении) может быть много различных файлов и каталогов. В этом смысле каталог выполняет две важные функции: в каталоге хранятся файлы и каталог соединяется с другими каталогами, как ветвь дерева соединяется с другими ветвями. Каждый каталог может содержать множество каталогов, но сам должен быть потомком только одного каталога.

При входе в систему пользователь оказывается в своей домашней директории, в которой он может создавать файлы, удалять их и копировать. Имя, которое дается каталогу или файлу при его создании, не является полным.

Полным именем является его путь. В ОС Windows путь состоит из имен всех каталогов, образующих путь к файлу от вершины дерева. Эти имена отделяются друг от друга обратной косой чертой (\ – обратный слэш). Путь дается и файлам. Фактически имя, которым система обозначает файл, представляет собой комбинацию имени файла с именами каталогов, образующих путь от корневого каталога до каталога данного файла.

Путь может быть абсолютным и относительным.

Абсолютный путь – это полное имя файла или каталога, начинающееся символом корневого каталога. Относительный путь начи-



нается символом текущего (рабочего) каталога и представляет собой обозначение пути к файлу относительно рабочего каталога.

Способ, которым данные организованы в байты, называется форматом файла. Для того чтобы прочесть файл, например, электронной таблицы, необходимо знать, каким образом байты представляют числа (формулы, текст) в каждой ячейке. Чтобы прочесть файл текстового редактора, надо знать, какие байты представляют символы, а какие шрифты или поля, а также другую информацию.

Все файлы условно можно разделить на две части – текстовые и двоичные.

Текстовые файлы – наиболее распространенный тип данных в компьютерном мире. Для хранения каждого символа чаще всего отводится один байт, а кодирование текстовых файлов выполняют с помощью специальных кодировочных таблиц.

Но чисто текстовые файлы встречаются все реже. Люди хотят, чтобы документы содержали рисунки, диаграммы и различные шрифты. В результате появляются форматы, представляющие различные комбинации текстовых, графических и других форм данных.

Двоичные файлы, в отличие от текстовых, не так просто просмотреть и в них, обычно, нет знакомых нам слов – лишь множество непонятных символов. Эти файлы не предназначены непосредственно для чтения. Примерами двоичных файлов являются исполняемые программы и файлы с графическими изображениями.

Каждый файл на диске имеет обозначение (полное имя), которое состоит из 2 частей: имени и расширения, разделенных точкой.

Расширение имени файла – необязательная последовательность символов, добавляемых к имени файла и предназначенных для идентификации типа (формата) файла. Это один из распространенных способов, с помощью которого пользователь или программное обеспечение компьютера может определить тип данных, хранящихся в файле.

В ранних операционных системах длина расширения была ограничена тремя символами, в современных операционных системах это ограничение отсутствует.

Операционная система или менеджер файлов могут устанавливать соответствия между расширениями файлов и приложениями.

Когда пользователь открывает файл с зарегистрированным расширением, автоматически запускается соответствующая этому расширению программа. Некоторые расширения показывают, что файл

сам является программой. Зачастую расширение файла отображается для пользователя пиктограммой (рис. 10.7).

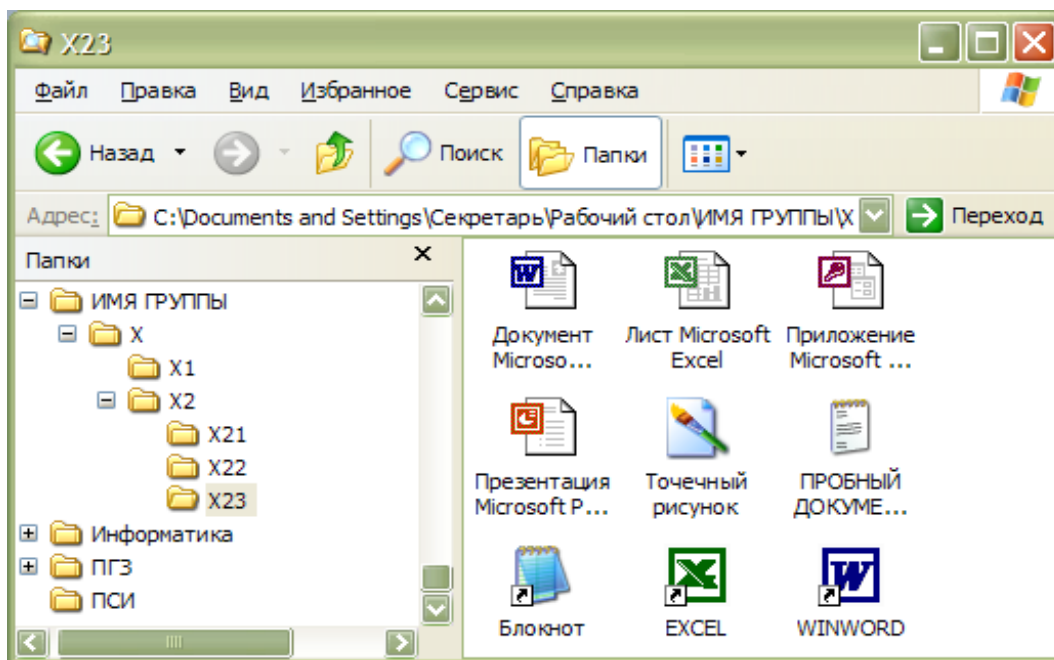


Рисунок 10.7 – Расширение файла в графическом интерфейсе ОС Windows отображается пиктограммой.

#### Наиболее часто встречающиеся расширения

Архивные файлы	.arj, .arc, .zip, .zoo, .pak, .lzh, .rar, .cab
Графические файлы	.bmp, .wmf, .pic, .wpg, .wpb, .tif, .gif, .icp, .pcx, .grf, .drw, .clp, .msp, .psd, .pcd
Файлы редакторов документов	.txt, .doc, .wps, .vw4, .wri, .sam, .jw, .ws7, .pdf, .html
Файлы табличных процессоров	.wks, .wki, .wk3, .wkt, .wxq, .wrk, .xls, .wqi, .vpp, .nod

### 10.4. Сервисное ПО (Служебные программы)

Сервисное ПО – это совокупность программных продуктов, предоставляющих пользователю дополнительные услуги в работе с компьютером и расширяющих возможности операционных систем.

По функциональным возможностям сервисные средства можно подразделить на средства:

- улучшающие пользовательский интерфейс;
- защищающие данные от разрушения и несанкционированного доступа;

- восстанавливающие данные;
- ускоряющие обмен данными между диском и ОЗУ;
- средства архивации и разархивации.
- антивирусные средства.

По способу организации и реализации сервисные средства могут быть представлены: оболочками, утилитами и автономными программами. Разница между оболочками и утилитами зачастую выражается лишь в универсальности первых и специализации вторых.

Оболочки, являющиеся надстройками над операционными системами (ОС), называются операционными оболочками. Оболочки являются настройками над операционной системой. Утилиты и автономные программы имеют узкоспециализированное назначение и выполняют каждая свою функцию. Но утилиты в отличие от автономных программ выполняются в среде соответствующих оболочек. При этом они конкурируют в своих функциях с программами ОС и другими утилитами. Поэтому классификация сервисных средств по их функциям и способам реализации является достаточно размытой и весьма условной. Оболочки предоставляют пользователю качественно новый интерфейс и освобождают его от детального знания операции и команд ОС.

Функции большинства оболочек, например семейства MS-DOS, направлены на работу с файлами и каталогами и обеспечивают быстрый поиск файлов; создание, просмотр и редактирование текстовых файлов; выдачу сведений о размещении файлов на дисках, о степени занятости дискового пространства и ОЗУ. Все оболочки обеспечивают ту или иную степень защиты от ошибок пользователя, что уменьшает вероятность случайного уничтожения слайдов.

Среди имеющихся оболочек для семейства MS-DOS наиболее популярна оболочка Norton Commander.

Утилиты предоставляют пользователю дополнительные услуги (не требующие разработки специальных программ) в основном по обслуживанию дисков и файловой системы. Эти утилиты чаще всего позволяют выполнять следующие функции:

- обслуживание дисков (форматирование, обеспечение сохранности информации, возможности ее восстановления в случае сбоя и т. д.);
- обслуживание файлов и каталогов (аналогично оболочкам);
- создание и обновление архивов;
- предоставление информации о ресурсах компьютера, о дисковом пространстве, о распределении ОЗУ между программами;

- печать текстовых и других файлов в различных режимах и форматах;
- защита от компьютерных вирусов.

Из утилит, получивших наибольшую известность, можно назвать многофункциональный комплекс Norton Utilities.

Под программами технического обслуживания понимается совокупность программно-аппаратных средств для диагностики и обнаружения ошибок в процессе работы компьютера или вычислительной системы в целом.

Они включают:

- средства диагностики и тестового контроля правильности работы ЭВМ и ее отдельных частей, в том числе автоматического поиска ошибок и неисправностей с определенной локализацией их в ЭВМ;
- специальные программы диагностики и контроля вычислительной среды информационной системы в целом, в том числе программно-аппаратный контроль, осуществляющий автоматическую проверку работоспособности системы обработки данных перед началом работы вычислительной системы в очередную производственную смену.

## **10.5. Компьютерные вирусы и антивирусные средства**

### ***10.5.1. Компьютерные вирусы***

Компьютерным вирусом называется программа, которая способна создавать свои копии (не обязательно полностью совпадающие с оригиналом) и внедрять их в различные объекты, ресурсы компьютерных систем и сетей без ведома пользователя. При этом копии сохраняют способность дальнейшего распространения.

Компьютерные вирусы, как и биологические, ставят перед собой три задачи: заразить, выполнить, размножиться. Заражается компьютер, когда человек запускает на исполнение некую программу, которая либо заражена вирусом (т. е. при ее выполнении запускается и вирус), либо сама является вирусом.

К сожалению, единственный действенный метод не заразить компьютер – не включать его вовсе. Можно еще посоветовать ничего не устанавливать и ничего не запускать. Только какой тогда смысл иметь компьютер?

Практически все вирусы функционируют в операционных системах семейства MS Windows и в MS DOS.

Сегодня 98 % всех вредителей попадает на компьютер через электронную почту.

В операционной системе Linux вирусы были выявлены только в лабораторных условиях. Несмотря на то, что некоторые образцы Linux-вирусов действительно обладали всеми необходимыми способностями к размножению и автономной жизни, ни один из них так и не был зафиксирован в «диком» виде. Использование ОС Linux защищает от вирусов гораздо лучше, чем любые антивирусные программы в MS Windows.

По способу распространения вредоносные программы можно условно разделить на *компьютерные вирусы, сетевые черви и троянские программы*.

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.

*Компьютерные вирусы* – это программы, способные размножаться самостоятельно, дописывая свой код к другим файлам или в служебные области диска. Каждый вирус способен выполнять деструктивные или нежелательные действия на зараженном компьютере. Вирус мешает другим программам и самому пользователю работать на компьютере.

Существует множество разновидностей вирусов. Самыми старыми являются файловые вирусы. Они размножаются, используя файловую систему. Почти столь же древними являются загрузочные вирусы. Они так названы потому, что заражают загрузочный сектор (*boot sector*) жесткого диска. Загрузочные вирусы замещают код программы, получающей управление при запуске системы. Таким образом, после перезагрузки системы управление передается вирусу. Сегодня загрузочные вирусы встречаются редко.

С середины 90-х гг. получили распространение макровирусы. Эти вредители представляют программу на макроязыке. Макроязык – это средство создания программ, которое поддерживается некоторой системой обработки данных (например, программами семейства Microsoft Office, такими как Microsoft Word и Excel, а также другими программами, например, продуктами компании «1С»). Для размно-

жения макровирусы используют встроенные возможности, например, текстового или табличного редактора. Таким способом эти вредители переносят себя из одного зараженного файла в другой.

*Сетевые черви.* «Червей» часто называют вирусами, хотя это не совсем верно. 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. Сетевые черви могут вообще не обращаться к ресурсам компьютера (за исключением оперативной памяти).

*Троянские программы,* «троянские кони» и просто «троянцы» – это вредоносные программы, которые сами не размножаются. Подобно знаменитому троянскому коню из «Илиады» Гомера, программа-троянец выдает себя за что-то полезное. Чаще всего троянец маскируется под новую версию бесплатной утилиты, какую-то популярную прикладную программу или игру. Таким способом троянец пытается заинтересовать пользователя и побудить его переписать и установить на свой компьютер вредителя самостоятельно.

По выполняемым вредоносным действиям троянские программы можно условно разделить на следующие виды:

1. Утилиты несанкционированного удаленного администрирования (позволяют злоумышленнику удаленно управлять зараженным компьютером).

2. Утилиты для проведения DDoS-атак (*Distributed Denial of Service* – распределенные атаки типа отказ в обслуживании).

3. Шпионские и рекламные программы, а также программы дозвона.

4. Серверы рассылки спама.

Многокомпонентные троянцы-загрузчики (переписывают из Интернета и внедряют в систему другие вредоносные коды или вредоносные дополнительные компоненты).

На практике часто встречаются программы-троянцы, относящиеся сразу к нескольким перечисленным выше видам.

***Утилиты несанкционированного удаленного администрирования***

Удаленное управление компьютером часто используется в крупных и средних компаниях, а также в тех случаях, когда необходимо оказать техническую помощь пользователю, находящемуся на значительном расстоянии. С помощью средств удаленного управления сис-

темный администратор может настроить каждый компьютер в организации, не вставая со своего рабочего места. Однако эта полезная функциональность в руках злоумышленника превращается в грозное оружие. «Троянские кони» часто представляют собой вполне легальные утилиты удаленного управления, адаптированные под нужды хакеров. Если злоумышленнику удастся внедрить такого троянца в чужую систему, он сможет незаметно управлять этим компьютером втайне от его настоящего владельца.

Управление зараженным компьютером обычно осуществляется через Интернет. Вот лишь небольшая часть того, что может сделать злоумышленник на инфицированном ПК: выкрасть любую информацию с компьютера-жертвы (файлы, пароли, реквизиты и т. д.), провести любую файловую операцию (отформатировать жесткий диск, стереть или переименовать какие-то файлы и т. д.), перезагрузить компьютер, подключиться к сетевым ресурсам, использовать зараженный компьютер для атаки на какой-то третий компьютер или сервер в Интернете.

### ***Утилиты для проведения DoS и DDoS-атак***

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 типа отказ в обслуживании 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.

Для того чтобы исчерпать ресурсы web-сервера, злоумышленник должен искусственно создать повышенную нагрузку на него. Каждый web-сервер тратит определенные ресурсы (память, вычислительные мощности и т. д.) на обработку входящих запросов. Если большое число компьютеров, на которых установлена утилита для проведения DDoS-атак, одновременно начнут посылать свои запросы web-серверу, то велика вероятность, что ресурсы web-сервера быстро исчерпаются, а сам сервер не сможет обслуживать легальных пользователей.

Технология DDoS-атак изначально была разработана отнюдь не для преступных целей. Напротив, она использовалась для тестирова-

ния пропускной способности каналов передачи данных и максимальной нагрузки, с которой узел сети может справиться.

### ***Шпионское и рекламное ПО, программы дозвона***

Шпионские программы втайне наблюдают за действиями пользователя и записывают в свой журнал интересующие злоумышленника события. Существует класс программ – клавиатурные шпионы. Эти вредители следят за пользователем и записывают каждое нажатие клавиши. По команде хакера или через определенное время клавиатурный шпион отправляет собранные сведения на компьютер злоумышленника. Существуют также троянцы-шпионы, которые отправляют на удаленный компьютер пароли и другую личную информацию пользователя.

Суть рекламных программ вытекает из их названия: эти компьютерные паразиты любым способом пытаются рекламировать продукты или услуги каких-то третьих компаний. Чтобы добиться своего, рекламные программы могут встроить рекламные объявления в какое-нибудь наиболее часто используемое приложение, например, в web-браузер. Рекламные программы также используют всплывающие окна, в которых либо показывается объявление, либо сразу же загружается рекламная страница из Интернета. Наиболее недобросовестным и трудноопределяемым способом рекламы является подтасовывание результатов поиска в Интернете. Когда пользователь ищет что-то в поисковой машине, рекламная программа изменяет результаты поиска таким способом, чтобы наверху оказались ее собственные рекламные ссылки и объявления.

Программы дозвона – это компьютерные паразиты, которые пытаются с помощью модема и телефонной линии дозвониться до платного сервера, находящегося чаще всего в другой стране. Такие серверы обычно предоставляют различные услуги порнографического характера и берут со своих пользователей поминутную оплату. Если вредоносной программе дозвона удастся соединиться с таким удаленным сервером и продержат соединение несколько минут, то пользователю потом придет счет на оплату услуг порнографического сервера и международные переговоры.

### **Серверы рассылки спама**

Спам, или нежелательные электронные сообщения, будет подробно рассмотрен дальше. Здесь мы остановимся лишь на средствах рассылки непрошенных сообщений.



Чтобы избежать ответственности за рассылку спама, злоумышленники не рассылают письма со своего компьютера. Они предпочитают заразить компьютеры других пользователей Интернета специальным троянцем, который превратит чужой ПК в сервер рассылки спама. Злоумышленнику останется лишь указать троянской программе, какое письмо и по каким адресам следует рассылать. Ответственность за эти незаконные действия будет нести легальный пользователь зараженного компьютера.

### ***Основные признаки появления в системе вируса***

При заражении компьютера вирусом важно его обнаружить. Для этого следует знать об основных признаках проявления вирусов. К ним можно отнести следующие:

- прекращение работы или неправильная работа ранее успешно функционировавших программ;
- медленная работа компьютера;
- невозможность загрузки операционной системы;
- исчезновение файлов и каталогов или искажение их содержимого;
- изменение даты и времени модификации файлов;
- изменение размеров файлов;
- неожиданное значительное увеличение количества файлов на диске;
- существенное уменьшение размера свободной оперативной памяти;
- вывод на экран непредусмотренных сообщений или изображений;
- подача непредусмотренных звуковых сигналов;
- частые зависания и сбои в работе компьютера.
- замедление работы некоторых программ;
- заметное снижение скорости работы в Интернете (вирус могут передавать информацию по сети).

Следует отметить, что вышперечисленные явления не обязательно вызываются присутствием вируса, а могут быть следствием других причин. Поэтому без специальных мероприятий и антивирусных средств затруднена правильная диагностика состояния компьютера.

## ***10.5.2. Антивирусные мероприятия***

### *Административные антивирусные мероприятия*

Говоря об антивирусной защите, требуется разделять корпоративные и частные системы. Если речь идет об информационной безопасности организации, то необходимо позаботиться не только о технических (программных и аппаратных) средствах, но и об административных.

Если в некоторой компании есть сеть, не связанная с Интернетом, то вирус извне туда не проникнет, а чтобы вирус случайно не попал в корпоративную сеть изнутри, можно просто не давать пользователям возможности самостоятельно считывать носители информации, такие как CD-диски, USB-флэш или выходящие из употребления дискеты. Например, если кому-то из сотрудников необходимо считать что-либо с CD, он должен обратиться к администратору, который имеет право установить CD и считать данные. При этом за проникновение вирусов с этого CD уже несет ответственность администратор.

При нормальной организации безопасности в офисе именно администратор контролирует установку любого ПО; там же, где сотрудники бесконтрольно устанавливают софт, в сети рано или поздно появляются вирусы.

Большинство случаев проникновения вирусов в корпоративную сеть связано с выходом в Интернет с рабочей станции. Существуют режимные организации, где доступ к Интернету имеют только неподключенные к корпоративной сети станции. В коммерческих организациях такая система неоправданна. Там Интернет-канал защищается межсетевым экраном и прокси-сервером. Во многих организациях разрабатывается политика, при которой пользователи имеют доступ лишь к тем ресурсам Интернета, которые нужны им для работы.

Конечно, поддержка политики жесткого разграничения прав доступа требует дополнительных инвестиций, а в ряде случаев приводит к замедлению выполнения некоторых работ. Поэтому каждая компания должна искать для себя разумный компромисс, сопоставляя финансовые потери от порчи информации и замедления бизнес-процессов. В ситуации, когда документы содержат важные стратегические данные, например, государственную тайну, именно степень

ущерба в случае разглашения тайны определяет бюджет на меры безопасности.

Помимо антивирусной защиты, важно не забывать о таком важном средстве защиты данных, как резервное копирование. Резервное копирование является стратегическим компонентом защиты данных. Если данные уничтожены вирусом, но у администратора есть вовремя сделанная резервная копия, потери будут минимальными.

### ***Краткий обзор антивирусных пакетов***

**AVP** (*Antiviral Toolkit Pro, Kaspersky Antivirus, [www.kaspersky.ru](http://www.kaspersky.ru)*) – с помощью программ лаборатории Касперского пресекаются все возможные пути проникновения вирусов в компьютер пользователя, включая Интернет, электронную почту, дискеты и т. д. Здесь используются все типы антивирусной защиты – сканеры (AVP сканер), мониторы (AVP монитор), поведенческие блокираторы и ревизоры изменений.

**Dr. WEB** (*[www.drweb.ru](http://www.drweb.ru)*) – не менее известный пакет от лаборатории Данилова. Это в первую очередь программа-полифаг, предназначенная для поиска и обезвреживания файловых, загрузочных и файлово-загрузочных вирусов. Ее отличает эвристический анализ (позволяет обнаружить вирусы, не известные ранее). Другие типы антивирусных средств (монитор, модуль для электронной почты) реализованы в виде отдельных программ.

**Norton Antivirus** (*[www.symantec.com](http://www.symantec.com)*) – мощнейший антивирусный пакет фирмы Symantec, включающий в себя все типы антивирусных средств. Особенно надежную защиту пакет представляет от сетевых вирусов. Также достаточно хорошо развиты эвристические возможности программы и удобный и быстрый способ обновления антивирусных баз.

**NOD32** (*[www.NOD32.ru](http://www.NOD32.ru)*) – антивирусный пакет, распространяемый в режиме свободного ПО. Он имеет достаточно полный набор антивирусных модулей и по своим возможностям сравним с вышеперечисленными антивирусными программами.

Совершенный механизм обновлений антивирусных баз значительно укрепляет защиту персонального компьютера. Пользователи антивирусных программ могут получать круглосуточную техническую поддержку на сайте производителя.

## 10.6. Инструментальные системы программирования

Комплекс средств, включающих входной язык программирования, транслятор, машинный язык, библиотеки стандартных программ, средства отладки оттранслированных программ и компоновки их в единое целое, называется **системой программирования**.

Входной язык программирования называется языком высокого уровня по отношению к машинному языку, называемому языком низкого уровня.

Особое место в системе программирования занимают ассемблеры, представляющие собой комплекс, состоящий из входного языка программирования ассемблера и ассемблер-компилятора. **Ассемблер** представляет собой мнемоническую (условную) запись машинных команд и позволяет получить высокоэффективные программы на машинном языке. Однако его использование требует высокой квалификации программиста и больших затрат времени на составление и отладку программ.

Наиболее распространенными языками программирования являются: Pascal, Basic, C++, Fortran и др. Тенденции развития – появление языков четвертого поколения типа Visual Basic.

**Транслятором** языка программирования называется программа, осуществляющая перевод текста программы с языка программирования в (как правило) машинный код.

В системе программирования транслятор переводит программу, написанную на входном языке программирования, на язык машинных команд конкретной ЭВМ. В зависимости от способа перевода с входного языка (языка программирования) трансляторы подразделяют на **компиляторы и интерпретаторы**.

В компиляции процессы трансляции и выполнения программы разделены во времени. Сначала компилируемая программа преобразуется в набор объектных модулей на машинном языке, которые затем собираются (компонуются) в единую машинную программу, готовую к выполнению и сохраняемую в виде файла на магнитном диске. Эта программа может быть выполнена многократно без повторной трансляции.

**Интерпретатор** осуществляет пошаговую трансляцию и немедленное выполнение операторов исходной программы: каждый оператор входного языка программирования транслируется в одну или не-

сколько команд машинного языка, которые тут же выполняются без сохранения на диске. Таким образом, при интерпретации программа на машинном языке не сохраняется и поэтому при каждом запуске исходной программы на выполнение ее нужно (пошагово) транслировать заново. Главным достоинством интерпретатора по сравнению с компилятором является простота.

**Библиотека стандартных программ** – организованная на машинном носителе совокупность компьютерных программ, доступ к которым осуществляется по их именам (или индексам). Библиотека стандартных программ ускоряет и облегчает процесс создания пользовательской программы, позволяя применять в качестве составных частей уже готовые функции (подпрограммы), для чего достаточно указать имя соответствующей функции. Благодаря библиотеке стандартных программ процесс программирования становится более технологичным, а сама пользовательская программа легче для восприятия.

Существенной частью программного обеспечения библиотеки стандартных программ является набор **сервисных программ (утилит)**, обеспечивающих создание библиотеки, включение и исключение функций (подпрограмм), вывод оглавления и др.

## 10.7. Драйверы

С компьютером могут сопрягаться разнообразные устройства: видеокарта, звуковая карта, принтер, сканер, манипуляторы, дисководы, цифровые фотоаппараты, сотовые телефоны. Каждое из них имеет свой набор команд – свой язык.

Чтобы конкретная операционная система могла управлять конкретным устройством, прибегают к помощи программ-переводчиков, знающих с одной стороны язык команд конкретного устройства, а с другой – язык конкретной операционной системы, под управлением которой должно работать это устройство.

Такая программа называется **драйвером** (*driver*) и поставляется вместе с устройством его производителем. Производители аппаратного ПО, как правило, также размещают драйверы, созданных ими устройств, на своих web-сайтах.

## 10.8. Архиваторы

Архиватор – это программа, которая сжимает файл или группу файлов в один архивный файл с целью уменьшения их размера. При этом не теряется ни бита информации, и любой файл можно из архива извлечь.

Что дает архивация? Во-первых, экономию места на диске, во-вторых, возможность перенести большой объем информации на диске, в-третьих, есть возможность пересылать большие файлы по электронной почте.

### *Наиболее популярные архиваторы*

ZIP – еще со времен ОС DOS один из самых популярных и распространенных архивных форматов, основанный на алгоритмах сжатия, предложенных в 80-х гг. прошлого столетия израильскими математиками Лемпелем и Зивом. Он отличается приемлемой степенью сжатия информации и достаточно высоким быстродействием. Сегодня он является стандартом де-факто в Интернете, и его поддерживают практически все программы-архиваторы.

RAR – разработан российским программистом Евгением Рошалем и позволяет получить размер сжатого файла гораздо меньший, чем ZIP, ценой этому является более продолжительный процесс обработки архива. В целом формат RAR значительно лучше других оптимизирован для решения сложных задач с использованием большого количества файлов и гигабайтных дисковых пространств.

СAB – применяется в продуктах Microsoft как стандартный для упаковки файлов, причем его алгоритм, нигде не опубликованный, представляет собой достаточно совершенный продукт, имеющий высокий коэффициент сжатия.

GZIP, TAR – получили наибольшее распространение в системах на базе Unix и ее самой популярной разновидности Linux.

ACE – достаточно новый формат с высокой степенью сжатия, завоевывающий все большую популярность.

Архив, созданный тем или иным архиватором, имеет расширение, соответствующее названию архиватора. Например, расширение файла `archive.rar` говорит о том, что он был создан с помощью архиватора RAR.

Архиваторы различаются возможностями и качеством сжатия, которое зависит также и от типа сжимаемых данных. Некоторые архива-



качества может быть сжато даже в сотню раз, цветные рисунки – в пять-десять раз, а цветные изображения высокого фотографического качества сжимаются менее чем в два раза.

Хорошо сжимаются текстовые файлы (.TXT, .DOC, .PAS и др.). Если файл содержит текст, написанный на естественном языке, например, повесть или рассказ, то коэффициент сжатия будет равен двум-трем, а если в файле записан текст программы, то коэффициент сжатия может достигать пяти и более. Коэффициент сжатия исполняемых файлов (.EXE, и др.) также сильно колеблется, в среднем, он равен трем.

В отношении звука (.WAV, .AU), использование универсальных методов сжатия редко дает хорошие результаты: звуковой файл сокращается всего на 20–40 %.

То же самое касается и высококачественных изображений, имеющих много деталей. Поэтому для этих типов изображений используют специальные методы сжатия, особо хорошие результаты дают так называемые алгоритмы сжатия с потерями.

Одна из идей этих алгоритмов состоит в том, что человеческий глаз и ухо не очень восприимчивы к некоторым мелким деталям изображения или звука, поэтому лишнюю информацию можно просто отбросить. Разумеется, кроме отбрасывания лишней информации, используют и другие алгоритмы, в результате чего достигается несравнимый коэффициент сжатия при минимальных потерях качества (с точки зрения восприятия человека).

Такой подход используют в файлах формата JPEG, предназначенного для хранения статичных изображений и MPEG, предназначенного для хранения видео- и аудиоинформации. Сегодня особенно популярны такие форматы, как MPEG-3 (или MP3), предназначенный для хранения аудиоинформации и позволяющий достигнуть десятикратного сжатия почти без потери качества, и MPEG-4, используемый для хранения видеофильмов.

Плохо архивируются, либо вообще увеличиваются в размере файлы, данные в которых уже сжаты, в том числе архивы (.RAR, .ZIP, .ARJ и др.), графические файлы тех форматов, которые имеют собственное сжатие (.GIF, .JPG, .PNG и др.), аудиофайлы (.MP3), видеофайлы (.MPG, .AVI и др.), самораспаковывающиеся архивы (.EXE).



## 10.9. Программы обслуживания жестких дисков

Основные операции, которые необходимо иногда проводить с жесткими дисками:

- **Разбиение на разделы.** На жесткий диск может быть установлено одновременно несколько операционных систем. Для этого жесткий диск должен быть разбит на разделы, т. е. независимые области на диске, в каждом может быть создана своя файловая система. Наиболее простой и традиционно используемой программой для этих целей в Windows является программа FDisk. ОС Windows 2000, XP, Vista имеют встроенную программу разбиения жестких дисков на разделы.

- **Форматирование.** Оно делится на *низкоуровневое* (физическое) форматирование, которое выполняется производителями и делит поверхности магнитных пластин на дорожки и сектора и *высокоуровневое* (логическое) форматирование, которое заключается в разбиении на кластеры и размещении на диске файловой системы. Логическое форматирование выполняется стандартной программой ОС Windows Format (Форматирование дисков).

- **Проверка диска на наличие логических и физических ошибок.** Если каким-то образом соответствие между тем, что записано в загрузочной области диска, и тем, что на самом деле находится на диске, нарушено, последствия могут быть непредсказуемы. Это может возникнуть вследствие сбоя ОС и другого ПО. В частности, велика вероятность возникновения ошибок при некорректном завершении работы компьютера, при зависании системы и т. д. Обнаружить возникшие проблемы и предотвратить неприятности поможет стандартная программа Windows Проверка диска или ScanDisk. Но эта программа недостаточно мощна и функциональна. Поэтому при серьезных проблемах необходимо использовать более мощные средства (например, Norton Disk Doctor (NDD) из пакета Norton Utilities фирмы Symantec).

- **Дефрагментация.** Как известно, с точки зрения быстродействия, винчестер одно из самых слабых мест системы. Каждый файл на диске занимает определенное пространство. Это пространство разбито на блоки – кластеры. Каждый кластер принадлежит определенному файлу. Хорошо, если кластеры одного файла следуют подряд, но так бывает не всегда. Файлы на диске постоянно создаются и уничтожаются. Операционная система не всегда может выделить файлу место таким образом, чтобы его кластеры шли друг за другом. То

есть файл может занимать несколько кластеров, разбросанных по разным местам диска. В этом случае говорят, что файл фрагментирован. При этом скорость чтения и записи файла замедляется заметно. Если на диске образуется много таких файлов, то скорость работы системы заметно падает. Для решения этой проблемы помогает стандартная программа Windows Дефрагментация диска или Defrag. Опять же можно порекомендовать использовать более мощное средство дефрагментации (например, Norton Speed Disk из Norton Utilities).

- **Очистка диска.** При регулярной работе на компьютере иногда накапливается некоторый пользовательский и системный «мусор», который полезно периодически расчищать и ликвидировать. Для этого существует много различных программ, а в Windows существует утилита – Очистка диска.

## **10.10. Прикладное программное обеспечение**

Прикладные программы (приложения, application) предназначены для того, чтобы обеспечить применение вычислительной техники в различных сферах деятельности человека.

Из-за огромного разнообразия прикладного программного обеспечения (ППО) существует множество вариантов его классификации. Наиболее часто их разделяют на два основных класса:

1. **Прикладные программы общего назначения.** К таким относятся программы, обеспечивающие выполнение наиболее часто используемых, универсальных задач (текстовые редакторы, табличные процессоры, графические редакторы, Интернет-клиенты и т. п.).

2. **Прикладные программы специального (профессионального) назначения.** Программы этого класса ориентированы на достаточно узкую предметную область, (издательские системы; САПР – системы автоматизированного проектирования; банковские, бухгалтерские программы; программы 3D-графики; программы видеомонтажа; нотные редакторы и т. д.).

### ***10.10.1. Средства обработки текстовой информации***

Несмотря на широкие возможности использования компьютеров для обработки разной информации, самыми популярными по-прежнему остаются программы, предназначенные для работы с текстом. При подготовке текстовых документов на компьютере используются три основные группы операций:

1. Операции ввода позволяют перенести исходный текст из его внешней формы в электронный вид, т. е. в файл, хранящийся на компьютере. Ввод может осуществляться не только набором с помощью клавиатуры, но и путем сканирования бумажного оригинала и последующего перевода документа из графического формата в текстовый (распознавание).

2. Операции редактирования (правки) позволяют изменить уже существующий электронный документ путем добавления или удаления его фрагментов, перестановки частей документа, слияния нескольких файлов, разбиения единого документа на несколько более мелких и т. д.

Ввод и редактирование при работе над текстом часто выполняются параллельно. При вводе и редактировании формируется содержание текстового документа.

3. Оформление документа задают операциями форматирования. Команды форматирования позволяют точно определить, как будет выглядеть текст на экране монитора или на бумаге после печати на принтере.

Программы, предназначенные для обработки текстовой информации, называют **текстовыми редакторами (процессорами)**.

Все многообразие современных текстовых редакторов условно можно разбить на три основные группы:

1. *Простейшие* текстовые редакторы, обладающие минимумом возможностей и способные работать с документами в обычном текстовом формате .txt. К этой группе редакторов можно отнести как входящие в комплект поставки ОС семейства Windows редакторы WordPad и совсем малофункциональный NotePad (Блокнот), и множество аналогичных продуктов других производителей (Atlantis, EditPad, Aditor Pro, Gedit и т. д.).

2. *Промежуточный* класс текстовых редакторов включает в себя достаточно широкие возможности по части оформления документов. Они работают со всеми стандартными текстовыми файлами (ТХТ, RTF, DOC). К таким программам можно отнести Microsoft Works, Лексикон под управлением Windows.

3. *Мощные текстовые процессоры*, такие, как Microsoft Word или StarOffice Writer. Они выполняют практически все операции с текстом. Большинство пользователей использует именно эти редакторы в повседневной работе.

*Основными функциями* текстовых редакторов (процессоров) являются:

- ввод и редактирование символов текста;
- возможность использования различных шрифтов символов;
- копирование и перенос части текста с одного места на другое или из одного документа в другой;
- контекстный поиск и замена частей текста;
- задание произвольных параметров абзацев и шрифтов;
- автоматический перенос слов на новую строку;
- автоматическую нумерацию страниц;
- обработка и нумерация сносок;
- создание таблиц и построение диаграмм;
- проверка правописания слов и подбор синонимов;
- построение оглавлений и предметных указателей;
- распечатка подготовленного текста на принтере и т. п.

Также практически все текстовые процессоры обладают следующими функциями:

- поддержка различных форматов документов;
- многооконность, т. е. возможность работы с несколькими документами одновременно;
- вставка и редактирование формул;
- автоматическое сохранение редактируемого документа;
- работа с многоколоночным текстом;
- возможность работы с различными стилями форматирования;
- создание шаблонов документов;
- анализ статистической информации.

Сегодня практически все мощные текстовые редакторы входят в состав интегрированных программных пакетов, предназначенных для нужд современного офиса. Так, Microsoft Word входит в состав самого популярного офисного пакета Microsoft Office.

Аналогичные MS Office программы – OpenOffice.org Writer, StarOffice Writer, Corel WordPerfect, Apple Pages.

### ***10.10.2. Средства обработки табличной информации***

Табличное представление данных имеет свои особенности. Многие виды данных намного удобнее хранить и обрабатывать в табличной форме, особенно числовые. Автоматизация табличных расчетов во много раз повышает эффективность и качество работы.

Компьютерные программы, предназначенные для хранения и обработки данных, представленных в табличном виде, называют **электронными таблицами или табличными процессорами**.

Первая электронная таблица VisiCalc была выпущена фирмой Visi Corporation в 1981 г., и именно с этого момента принято вести отсчет истории электронных таблиц как самостоятельного вида программного обеспечения.

Идея выделения таблиц в особый класс документов и создание специализированной программы, выполняющей всевозможные операции с табличными данными, оказалась весьма удачной и была подхвачена многими фирмами. Популярность электронных таблиц стремительно росла.

В 1983 г. фирма Lotus Development Corporation выпустила электронную таблицу 1-2-3, ставшую на долгие годы фактическим стандартом в своей области.

В 1985 г. появилась первая наиболее распространенный на сегодня табличный процессор Microsoft Excel. Спустя год данный сектор desktop-приложений пополнился пакетом Quattro, созданным компанией Borland International Corporation.

Электронные таблицы позволяют решать целый комплекс задач:

**Выполнение вычислений.** Издавна многие расчеты выполняются в табличной форме, особенно в области делопроизводства: многочисленные расчетные ведомости, табуляграммы, сметы расходов и т. п. Кроме того, решение численными методами целого ряда математических задач удобно выполнять в табличной форме. Электронные таблицы представляют собой удобный инструмент для автоматизации таких вычислений. Решения многих вычислительных задач на ЭВМ, которые раньше можно было осуществить только путем программирования, стало возможно реализовать на электронных таблицах.

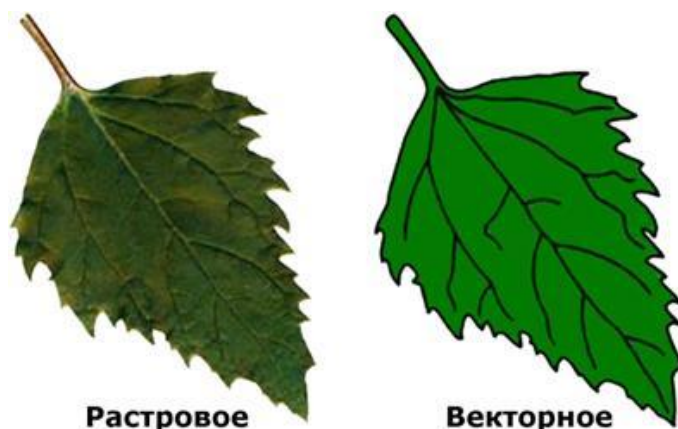
**Математическое моделирование.** Использование математических формул в ЭТ позволяет представить взаимосвязь между различными параметрами некоторой реальной системы. Основное свойство ЭТ – мгновенный пересчет формул при изменении значений входящих в них операндов. Благодаря этому свойству, таблица представляет собой удобный инструмент для организации эксперимента: подбор параметров, прогноз поведения моделируемой системы, анализ зависимостей, планирование. Дополнительные удобства для моделирования дает возможность графического представления данных.

**Использование электронной таблицы в качестве базы данных.** Конечно, по сравнению с СУБД электронные таблицы имеют меньшие возможности в этой области. Однако некоторые операции манипулирования данными, свойственные реляционным СУБД, в них реализованы. Это поиск данных по заданным условиям и сортировка данных.

Одним из самых популярных табличных процессоров сегодня является MS Excel, входящий в состав пакета Microsoft Office. Главные конкуренты – OpenOffice.org Calc, StarOffice Calc и Corel Quattro Pro.

### ***10.10.3. Средства обработки графической информации***

Для работы с компьютерной графикой существует множество классов ПО. Различают обычно два основных вида компьютерной графики, которые отличаются принципами формирования изображения: **растровая и векторная** (рис. 10.8). Ранее, в параграфе «Кодирование графических данных» приведены некоторые сведения об основах представления растровой и графической информации.



*Рисунок 10.8 – Сравнение растрового и векторного изображения*

#### ***Растровая графика***

Компьютерное растровое изображение представляется в виде прямоугольной матрицы, каждая ячейка которой – цветная точка – пиксель.

При создании растровых изображений необходимо задавать разрешение и размеры изображения.

Разрешение изображения измеряется в точках на дюйм (*dots per inch – dpi*) (1 дюйм = 25,4 мм). Полиграфическая печать полноцветного изображения требует разрешения не менее 200–300 dpi.

С помощью растровой графики можно отразить и передать всю гамму оттенков и тонких эффектов, присущих реальному изображению. Растровое изображение ближе к фотографии, оно позволяет более точно воспроизводить основные характеристики фотографии: освещенность, прозрачность и глубину резкости.

Чаще всего растровые изображения получают с помощью сканирования фотографий и других изображений, с помощью цифровой фотокамеры или путем захвата кадра видеосъемки.

Основным недостатком растровых изображений является невозможность их увеличения для рассмотрения деталей. При увеличении изображения точки становятся крупнее, но дополнительная информация не появляется. Этот эффект называется пикселизацией (см. рис. 2.1).

### ***Средства работы с растровой графикой***

К числу простейших растровых редакторов относятся PaintBrush, Paint, Painter, которые позволяют рисовать простейшие растровые изображения.

Основной класс растровых графических редакторов предназначен для обработки готовых растровых изображений с целью улучшения их качества и создания собственных изображений из уже имеющихся. К таким редакторам относятся такие мощные программы, как Adobe Photoshop, Corel PhotoPaint, Gimp и др.

### **Основные растровые форматы**

**BMP** (*Windows Device Independent Bitmap*) – самый простой растровый формат является форматом Windows, он поддерживается всеми графическими редакторами, работающими под ее управлением. В BMP данные о цвете хранятся только в модели RGB, поддерживаются как индексированные цвета (до 256 цветов), так и полноцветные изображения. Благодаря примитивнейшему алгоритму записи изображения, при обработке файлов формата BMP расходуется очень мало системных ресурсов, поэтому этот формат часто используется для хранения логотипов, экранных заставок, иконок и прочих элементов графического оформления программ.

**GIF** (*Graphics Interchange Format*) – является одним из самых популярных форматов изображений, размещаемых на веб-страницах. Отличительной его особенностью является использование режима индексированных цветов (не более 256), что ограничивает область применения формата изображениями, имеющими резкие цветовые переходы. Небольшие размеры файлов изображений обусловлены применением алгоритма сжатия без потерь качества, благодаря чему

изображения в этом формате наиболее удобны для пересылки по каналам связи глобальной сети. В GIF реализован эффект прозрачности и возможности хранить в одном файле несколько картинок с указанием времени показа каждой, что используется для создания анимированных изображений.

**PNG** (*Portable Network Graphics*) – формат, являющийся плодом трудов сообщества независимых программистов, появился на свет как ответная реакция на переход популярнейшего формата GIF в разряд коммерческих продуктов. Этот формат в отличие от GIF сжимает растровые изображения не только по горизонтали, но и по вертикали, что обеспечивает более высокую степень сжатия. Как недостаток формата часто упоминается то, что он не дает возможности создавать анимационные ролики. Зато формат PNG позволяет создавать изображения с 256 уровнями прозрачности что, безусловно, выделяет его на фоне всех существующих в данный момент форматов. Так как формат создавался для Интернета, в его заголовке не предназначено место для дополнительных параметров типа разрешения, поэтому для хранения изображений, подлежащих печати, PNG плохо подходит, для этих целей лучше подойдет PSD или TIFF.

**JPEG** (*Joint Photographic Experts Group*) – самый популярный формат для хранения фотографических изображений, является общепризнанным стандартом. JPEG может хранить только 24-битовые полноцветные изображения. Хотя JPEG отлично сжимает фотографии, но это сжатие происходит с потерями и портит качество, тем не менее, он может быть легко настроен на минимальные, практически незаметные для человеческого глаза, потери. Однако не стоит использовать формат JPEG для хранения изображений, подлежащих последующей обработке, так как при каждом сохранении документа в этом формате процесс ухудшения качества изображения носит лавинообразный характер. Наиболее целесообразно будет корректировать изображение в каком-нибудь другом подходящем формате, например, TIFF, и лишь по завершению всех работ окончательная версия может быть сохранена в JPEG. Формат JPEG не поддерживает анимацию или прозрачный цвет, и пригоден в подавляющем большинстве случаев только для публикации полноцветных изображений, типа фотографических, в Интернете.

**TIFF** (*Tag Image File Format*). Как универсальный формат для хранения растровых изображений, TIFF достаточно широко используется, в первую очередь, в издательских системах, требующих изо-



бражения наилучшего качества. Кстати, возможность записи изображений в формате TIFF является одним из признаков высокого класса современных цифровых фотокамер. В этом формате поддерживаются такие чисто профессиональные возможности, как обтравочные контуры, альфа-каналы, возможность сохранять несколько копий изображения с разным разрешением и даже включать в файл слои. Благодаря своей совместимости с большинством профессионального ПО для обработки изображений, формат TIFF очень удобен при переносе изображений между компьютерами различных типов.

**PSD** (*Adobe Photoshop*) – является стандартным форматом пакета Adobe Photoshop и отличается от большинства обычных растровых форматов возможностью хранения слоев (*layers*). Он содержит много дополнительных переменных (не уступает TIFF по их количеству) и сжимает изображения иногда даже сильнее, чем PNG (в тех случаях, когда размеры файла измеряются не в килобайтах, а в десятках или даже сотнях мегабайт). Файлы PSD свободно читаются большинством просмотрщиков.

### ***Векторная графика***

Основным логическим элементом векторной графики является **геометрический объект**. В качестве объекта принимаются простые геометрические фигуры, так называемые **примитивы** – прямоугольник, окружность, эллипс, линия (рис. 10.9).

Благодаря этому форму, цвет и пространственное положение составляющих изображение объектов можно описывать с помощью **математических формул**.



*Рисунок 10.9 – Геометрические примитивы*

### ***Преимущества векторной графики:***

– экономна в плане дискового пространства. Это связано с тем, что сохраняется не само изображение, а только некоторые основные данные, используя которые, программа всякий раз воссоздает изображение заново;

– объекты векторной графики легко трансформируются и масштабируются, что не оказывает практически никакого влияния на качество изображения. Масштабирование, поворот, искривление сводятся к элементарным преобразованиям над векторами;

– программы векторной графики имеют развитые средства интеграции изображений и текста, единый подход к ним. Поэтому программы векторной графики незаменимы в области дизайна, технического рисования, для чертежно-графических и оформительских работ.

*Недостатки* векторной графики:

– ограничена в живописных средствах: в программах векторной графики практически невозможно создавать фотореалистичные изображения;

– векторный принцип описания изображения не позволяет автоматизировать ввод графической информации, как это делает сканер для растровой графики.

### **Средства работы с векторной графикой**

Как и в случае с растровой графикой, для работы с векторной имеется огромное количество программных средств, освоение которых сложнее по сравнению с растровыми.

К основным относят программы:

– CorelDraw – это профессиональный графический редактор с богатыми настройками и развитой системой управления;

– Adobe Illustrator – основное достоинство программы в том, что она вместе с Adobe Photoshop и Adobe PageMaker образует достаточно мощный пакет для выполнения компьютерной верстки полиграфических изданий и разработки сложных документов;

– Macromedia Freehand – один из самых дружелюбных и интуитивно понятных векторных редакторов. Программа отличается простотой системы управления и высоким быстродействием, но ее возможности несколько скромнее, чем у предыдущих редакторов.

### **Основные векторные графические форматы**

Своего рода стандартом стали форматы двух наиболее популярных профессиональных графических пакетов – Adobe Illustrator и CorelDRAW:

– **AI** (*Adobe Illustrator Document*) – поддерживают практически все программы, так или иначе связанные с векторной графикой. Этот формат является наилучшим посредником при передаче изображений из одной программы в другую, с PC на Macintosh и наоборот. В целом, несколько уступая CorelDRAW по иллюстративным возможно-

стям (может содержать в одном файле только одну страницу, имеет маленькое рабочее поле – этот параметр очень важен для наружной рекламы – всего 3×3 метра), тем не менее, он отличается наибольшей стабильностью и совместимостью с языком PostScript, на который ориентируются практически все издательско-полиграфические приложения.

– **CDR** (*CorelDRAW Document*) – основной рабочий формат популярного пакета CorelDRAW, являющимся неоспоримым лидером в классе векторных графических редакторов на платформе PC. Имея сравнительно невысокую устойчивость и проблемы с совместимостью файлов разных версий формата, тем не менее, формат CDR можно без натяжек назвать профессиональным. В файлах этих версий применяется отдельная компрессия для векторных и растровых изображений, могут внедряться шрифты, файлы CDR имеют огромное рабочее поле 45x45 метров, поддерживается многостраничность.

– **WMF** (*Windows Metafile*) – еще один формат Windows, на сей раз векторный. Понимается практически всеми программами Windows, так или иначе связанными с векторной графикой. Однако, несмотря на кажущуюся простоту и универсальность, пользоваться форматом WMF стоит только в крайних случаях, поскольку он не может сохранять некоторые параметры, которые могут быть присвоены объектам в различных векторных редакторах, не воспринимается Macintosh и способен исказить цветовую схему изображения.

– **PDF** (*Portable Document Format*) – первоначально проектировался как компактный формат электронной документации, но в последнее время все больше используется для передачи по сетям графических изображений и смешанных документов, содержащих как текст, так и графику. Формат PDF является в полной мере платформонезависимым форматом, в текстовой части которого возможно использование множества шрифтов (которые содержатся непосредственно в документе, поэтому документ будет выглядеть так, как задумал его автор, на любом компьютере) и гипертекстовых ссылок, а также графические иллюстрации любого типа (векторные или растровые). Для достижения минимального размера PDF-файла используется компрессия, причем каждый вид объектов сжимается по наиболее выгодному для него алгоритму. Просматривать документы в формате PDF и распечатывать их на принтере можно с помощью утилиты Acrobat Reader, распространяемой компанией Adobe бесплатно.

Особый класс программ для работы с любыми видами изображений представляют **программы-просмотрщики**. Они позволяют просматривать графические файлы различных форматов, создавать фотоальбомы на жестком диске, перемещать, переименовывать, изменять размеры, а также конвертировать из одного формата в другой изображения. Лидером в данной области является программа ACDSec.

#### ***10.10.4. 3D-графика***

Быстро развивается область трехмерной векторной (или 3D) графики.

К данному классу ПО относится, например, программа 3D-MAX Studio.

По сравнению с традиционными для графических программ двумерными векторными и растровыми объектами работа с 3D-графикой предполагает использование более сложных понятий и процедур, таких как сцена, камеры, источники света.

Возможности программ для работы с трехмерной графикой широки. Это и создание спецэффектов для кино и телевидения, получение реалистичных фотоизображений, технических иллюстраций в программах автоматизированного проектирования для разработки новых реальных объектов и т. д.

Использование 3D-программ напоминает съемку с помощью видеокамеры комнаты, полной сконструированных вами объектов. Они позволяют смоделировать комнату и ее содержимое с использованием разнообразных базовых трехмерных объектов (кубы, сферы, цилиндры, конусы). После того как модели всех объектов созданы и размещены на сцене, можно выбрать им оформление с помощью имеющихся в программе встроенных средств или создать собственное оформление. Затем можно создать и расставить воображаемые камеры, которые будут наблюдать и снимать виртуальный трехмерный мир. После всех приготовлений можно анимировать сцену, заставив двигаться объекты, источники света и камеры. В завершении можно визуализировать анимацию и зафиксировать результат в виде видеофайла.

### 10.10.5. Системы управления базами данных (СУБД)

**База данных** – совместно используемый набор логически связанных данных. Это единое хранилище данных, которое однократно определяется, а затем используется одновременно многими пользователями.

**Система управления базами данных (СУБД)** – это программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ.

В **реляционных** базах данных (БД самого распространенного типа) данные хранятся в **таблицах**. Столбцы называются **полями** (*fields*) и содержат данные определенного типа. Строки именуются **записями** (*records*). В одной строке хранится один набор данных, описывающих определенный объект.

Например, если в таблице хранятся данные о клиентах, она может содержать поля для имени, адреса, города, почтового индекса, номера телефона и т. д. Для каждого клиента будет создана отдельная запись.

Таблицы – не единственный тип объектов, из которых состоят базы данных. Помимо таблиц, существуют формы, отчеты и запросы.

**Формы** (*forms*) применяют для добавления новых данных и изменения уже существующих. Формы облегчают добавление и редактирование информации, а также позволяют контролировать тип вводимых данных и избегать при вводе ряда ошибок.

Для отображения данных в удобном для чтения виде используются **отчеты** (*reports*). Ознакомиться со всей информацией, хранящейся в таблице, сложно по той причине, что текст не умещается в полях целиком. Существует возможность включать в отчет не все данные, а только некоторые, что значительно повышает удобство использования.

Для вывода в отчеты определенных данных применяются **запросы** (*queries*). Использование запросов похоже на процесс поиска, – задаются конкретные критерии отбора, на основе которых база данных формирует и возвращает отчет. Например, если база данных содержит информацию о телефонных номерах, то можно запросить вывести в отчете только те телефоны, которые относятся к конкретному адресу, или только те, которые относятся к конкретной фамилии, или начинающиеся с определенных цифр и т. п. Запросы записываются на языке **SQL** (*Structured Query Language* – язык структурированных запросов).

В основе реляционных баз данных лежит **понятие связей** (отношений, *relationships*). Они позволяют разработчикам связывать несколько таблиц в базе посредством общих данных. При помощи взаимосвязей разработчики баз данных моделируют таблицы, отражающие взаимодействие объектов в реальности.

Возможности связей между таблицами не ограничены уменьшением избыточности данных. Они также позволяют создать SQL-запрос, извлекающий данные из нескольких таблиц на основе определенного критерия. Например, можно создать запрос, выводящий имена и фамилии всех клиентов, совершивших покупки на сумму, превышающую некоторое пороговое значение. Формы, предназначенные для записи сразу в несколько таблиц, тоже функционируют на основе связей.

Модель реляционной СУБД была разработана в 70–80 гг. XX века. К реляционным СУБД относится целый ряд программных продуктов, среди них Microsoft Access из пакета Microsoft Office, MySQL или более мощные системы промышленного уровня, таких как Microsoft SQL Server или Oracle.

В последнее время активно развивается и другая **модель представления баз данных – объектная**. Реляционная модель акцентирует свое внимание на структуре и связях сущностей, объектная – на их свойствах и поведении.

#### ***10.10.6. Средства разработки презентаций***

**Компьютерная презентация** представляет собой набор слайдов (электронных страниц), последовательность показа которых может меняться в процессе демонстрации презентации, т. е. презентация является **интерактивным** документом.

Презентация является **мультимедийным** документом, так как каждый слайд может включать в себя различные формы представления информации (текст, таблицы, диаграммы, изображения, звук, анимацию и др.).

Презентации обычно используют в процессе выступлений на конференциях, для рекламы товаров на выставках, при объяснении нового материала на уроке и т. д.

Наиболее популярной программой для создания презентаций является Microsoft Power Point. Основные конкуренты – OpenOffice.org Impress, StarOffice Impress, Corel WordPerfect и Apple Keynote.

### ***10.10.7. Автоматизация ввода информации в компьютер***

Основным методом перевода бумажных документов в электронную форму является **сканирование**. В результате сканирования получается графическое изображение, состоящее из точек, т. е. растровое изображение. Количество точек определяется как размером изображения, так и разрешением сканера.

Существуют специальные стандарты программного интерфейса, обеспечивающего связь между сканерами и операционной системой. Этот интерфейс основан на специальном протоколе TWAIN.

Графический образ, получаемый после сканирования документа, иногда необходимо перевести в текст. Для этого используют специальные программные средства, называемые **средствами распознавания образов**. Из программ, способных распознавать текст на русском языке, наиболее известной является ABBYY Fine Reader.

**Преобразование документа** в электронный вид происходит в **три этапа**. Каждый из этих этапов может выполняться программами как автоматически, так и под контролем пользователя.

**1. Сканирование.** Запускается сканирующий модуль, настраиваются параметры сканирования (разрешение, размер, тип сканирования) и происходит собственно сканирование.

**2. Сегментация и распознавание текста.** Прежде чем получить готовый текст, необходимо разбить фрагменты документа на блоки (текст, рисунок, таблица и т. д.), для того чтобы правильно их распознать (преобразовать в текстовый документ).

**3. Проверка орфографии и передача текстового документа в нужное приложение** для дальнейшей работы или сохранение в файл.

### ***10.10.8. Автоматизация перевода текста***

Идея автоматического перевода текстов с одного языка на другой зародилась с появлением первых компьютеров. Если бы полноценный перевод был возможен, то значительно упростилось бы общение между народами. Но это очень сложная задача, о полном решении которой пока говорить рано.

**Программы автоматического перевода** позволяют переводить отдельные слова и строить смысловые связи в предложениях, не все-

гда учитывая те или иные особенности языка. Поэтому они предназначены лишь для общего ознакомления с содержанием документа.

Программные *средства автоматического перевода* можно условно разделить на *две основные* категории:

**1. Компьютерные словари.** Их назначение – предоставить значения неизвестных слов быстро и удобно для пользователя.

**2. Системы автоматического перевода** позволяют выполнять автоматический перевод связного текста. В ходе работы программа использует словари и наборы грамматических правил, обеспечивающие наилучшее качество перевода.

### ***10.10.9. Издательские системы***

**Компьютерные издательские системы** – это комплекс аппаратных и программных средств, предназначенных для компьютерного набора, верстки и издания текстовых и иллюстративных материалов.

Главным отличием настольных издательских систем от текстовых редакторов является то, что они предназначены, в первую очередь, для оформления документов, а не для ввода и редактирования.

Процесс верстки состоит в оформлении текста и задании условий взаимного расположения текста и иллюстраций. Целью верстки является создание оригинал-макета, пригодного для размножения документа полиграфическим способом.

Существуют различные программные системы, среди которых можно выделить следующие:

1. *Adobe InDesign* – недавно появившийся пакет фирмы Adobe, оптимизированный под верстку документов самого широкого профиля, от одностраничных буклетов до толстых книг, обогащенный набором специфических визуальных инструментов.

2. *Adobe PageMaker* – еще один пакет фирмы Adobe, с довольно сложным интерфейсом и системой команд, но в то же время с высокой производительностью и богатыми возможностями, особенно при работе с цветом.

3. *Corel Ventura Publisher* – альтернативный пакет фирмы Corel, несколько утративший в последнее время свои позиции, но вследствие своей универсальности (имеет широкие функции обычных текстовых и графических редакторов, интеграция с Web, поддержка различных платформ) не потерявший актуальности.



4. *QuarkXPress* – достаточно легкая в освоении и гибкая издательская система, которая традиционно используется многими издательствами газет, журналов, рекламными агентствами.

### ***10.10.10. Системы автоматизации бухгалтерской деятельности***

К этому виду относят как простейшие программы для подготовки бухгалтерских документов и отчетности, так и очень сложные распределенные системы комплексного бухгалтерского учета 1С, Парус, Галактика, Инфо-бухгалтер и т. д.

**Программа «1С: Предприятие»** является универсальной системой автоматизации деятельности предприятия и может применяться на различных участках бухгалтерского учета.

Система содержит три основных компонента:

1. Бухгалтерский учет, отражающий финансовые операции.
2. Оперативный учет, предназначенный для автоматизированного учета движения и остатков товарных, материальных, денежных средств предприятия.
3. Расчет, позволяющий выполнять периодические расчеты различной сложности.

### ***10.10.11. Прочее программное обеспечение***

Можно перечислить еще множество прикладных программ как общего, как и специального назначения. В их числе:

- браузеры (*browsers*) – средства просмотра гипертекста, используемые для отображения информации, получаемой с веб-сайтов: MS Internet Explorer, FireFox, Opera и др.;
- клиенты электронной почты – MS Outlook Express, The Bat;
- аудио / видео проигрыватели – Winamp, DivX Player, Real Player, Windows Media Player;
- справочно-правовые системы – Консультант Плюс, Гарант, Кодекс;
- офис-менеджеры (органайзеры) – MS Outlook;
- ПО для инженерных расчетов – AutoCAD.

### **Вопросы для повторения и самоконтроля**

1. Дайте понятие термину программное обеспечение.
2. Что входит в состав программного обеспечения?

3. Что такое операционная система? Из каких программ она состоит?
4. Назначение операционных систем.
5. Интерфейс. Какие виды интерфейса вам известны?
6. Открытое и закрытое программное обеспечение. Почему оно так называется?
7. Что такое проприетарное программное обеспечение?
8. В чем заключается суть проекта GNU? Какие права имеет пользователь программного обеспечения в его среде? Что не имеет права делать пользователь программного обеспечения GNU?
9. Что такое программное обеспечение с открытым исходным кодом?
10. Есть ли различия между движениями открытого и свободного программного обеспечения?
11. Назовите наиболее распространенные семейства операционных систем.
12. Какие программы входят в базовую комплектацию операционной системы MS Windows и какие прикладные программы ее дополняют?
13. Какие программы входят в базовую комплектацию операционной системы Linux? Какие прикладные программы ее дополняют?
14. Что такое файловая система?
15. Дайте понятие файла, каталога, иерархической структуры каталогов.
16. В чем различие между абсолютным и относительным путем обращения к каталогу или файлу?
17. Что такое текстовые и двоичные файлы?
18. Из чего состоит полное имя файла?
19. Как реализуется соответствие между расширением файлов и приложениями?
20. Как по пиктограмме узнать тип файла?
21. Что такое сервисное (служебное) программное обеспечение?
22. Дайте понятие операционной оболочки. Приведите примеры наиболее распространенных операционных оболочек.
23. Дайте понятие утилиты. Приведите примеры наиболее распространенных утилит.
24. Что такое программы технического обслуживания?
25. Что подразумевается под компьютерным вирусом?

26. В каких операционных системах преобладают компьютерные вирусы?
27. На какие основные группы подразделяют компьютерные вирусы?
28. Какие типы вирусов вам известны?
29. Что такое сетевые черви?
30. Что такое троянские программы?
31. Расскажите о принципе действия утилит несанкционированного удаленного администрирования.
32. Расскажите о принципе действия DoS- и DDos-атак.
33. Что такое шпионское и рекламное программное обеспечение, программы дозвона?
34. Перечислите основные признаки появления в системе компьютерного вируса.
35. Из чего состоят административные антивирусные мероприятия?
36. Какие основные антивирусные пакеты применяются в России?
37. Что такое система программирования? Из чего она состоит?
38. Языки программирования высокого и низкого уровня?
39. Что такое транслятор? Какие два основных типа трансляторов существуют?
40. Назначение библиотеки стандартных программ.
41. Назначение программ-драйверов.
42. Для каких целей используют программы-архиваторы?
43. Перечислите наиболее популярные архиваторы, применяемые в России?
44. Что такое многотомные и самораспаковывающиеся архиваторы?
45. Приведите примеры способов сжатия информации архиваторами.
46. Файлы каких типов сжимаются лучше?
47. Что такое алгоритмы сжатия с потерями?
48. Что делают программы разбиения диска на разделы?
49. Что делают программы форматирования дисков?
50. Что делают программы проверки диска на наличие логических и физических ошибок?
51. Что делают программы дефрагментации диска?
52. Что делают программы очистки диска?
53. Назначение и основные классы прикладных программ.
54. Назначение и основные операции, выполняемые средствами обработки текстовой информации.

55. На какие группы можно разделить современные текстовые редакторы?
56. Каковы основные функции текстовых редакторов?
57. Какие электронные таблицы (табличные процессоры) оставили наибольший след в практике коммерческих вычислений?
58. Какие классы задач можно выполнять с помощью электронных таблиц?
59. В чем различие растровых и векторных графических пакетов? В чем их достоинства и недостатки?
60. Основные средства работы с растровой графикой.
61. Основные средства работы с векторной графикой.
62. Основные средства работы с трехмерной (3D) графикой.
63. Назначение систем управления базами данных. Какие основные объекты используют в них?
64. Что такое связи (отношения) в реляционных базах данных?
65. Назначение средств разработки презентаций.
66. Автоматизация ввода информации в компьютер с помощью сканеров.
67. Автоматизация перевода текстов.
68. Назначение издательских систем. Основные издательские системы.
69. Системы автоматизации бухгалтерской деятельности.
70. Какие программы можно отнести к прочему программному обеспечению?

### **Задания для самостоятельной работы**

1. В основные функции операционной системы не входит:
- а) обеспечение диалога с пользователем;
  - б) разработка программ для ЭВМ;
  - в) управление ресурсами компьютера;
  - г) организация файловой структуры;
  - д) обслуживание файловой структуры.
2. Драйверы – это:
- а) технические устройства;
  - б) системы автоматизированного проектирования;
  - в) программы для согласования работы внешних устройств и компьютера.
3. Установите соответствие между типами файлов и компонентами MS Office:

- 1) .xls;            а) презентация MS PowerPoint;
- 2) .doc;           б) текстовый документ MS Word;
- 3) .ppt;           в) база данных СУБД MS Access;
- 4) .mbd;           г) электронная таблица MS Excel.

4. К какому виду программного обеспечения относят Unix, DOS, Windows, Linux?

- а) операционные системы;
- б) информационные системы;
- в) инструментальное ПО.

5. Заражение компьютерными вирусами может произойти в процессе:

- а) форматирования дискеты;
- б) выключения компьютера;
- в) работы с файлами;
- г) печати на принтере.

6. На жестком диске C: в каталоге Robot хранится документ «resumeis.doc». Каков полный путь к файлу?

- а) Robot\resumeis.doc;
- б) c:\...\resumeis.doc;
- в) c:\Robot\resumeis.doc.

7. Укажите наиболее точное определение. Приложения – это:

- а) программы для проверки компьютерной системы;
- б) программы для ознакомления пользователя с принципами устройства компьютера;
- в) прикладные программы, созданные для работы под управлением конкретной операционной системы;
- г) техническая документация компьютера.

8. Установите соответствие между названием и функцией популярных программ:

- 1) КонсультантПлюс;            а) система распознавания текста;
- 2) MsSQL;                        б) справочно-правовая система;
- 3) Adobe Photoshop;        в) графический редактор;
- 4) Fine Reader;                г) СУБД.

9. Пикселизация изображений при увеличении масштаба – один из недостатков:

- а) растровой графики;
- б) векторной графики.

10. Одним из основных преимуществ растровой графики перед векторной является:

- а) возможность изменения разрешения изображения;

- б) возможность интеграции текста;
- в) фотореалистичность изображений;
- г) возможность трансформации изображения;
- д) малый размер изображения.

11. Разрешение изображения измеряется:

- а) в пикселях (рiх);
- б) точках на дюйм (dpi);
- в) мм, см, дюймах.

12. Минимальным объектом изображения в растровом графическом редакторе является:

- а) палитра цветов;
- б) знакоместо (символ);
- в) точка экрана (пиксель);
- г) объект (прямоугольник, круг и т.д.)

### *Ответы к заданиям для самостоятельной работы*

1. Верные ответы: б), г).

2. Верные ответы: в).

3. Верные ответы:

- |          |                                  |
|----------|----------------------------------|
| 1) .xls; | г) электронная таблица MS Excel; |
| 2) .doc; | б) текстовый документ MS Word;   |
| 3) .ppt; | а) презентация MS PowerPoint;    |
| 4) .mbd; | в) база данных СУБД MS Access.   |

4. Верные ответы: а).

5. Верные ответы: в).

6. Верные ответы: в).

7. Верные ответы: в).

8. Верные ответы:

- |                     |                                  |
|---------------------|----------------------------------|
| 1) КонсультантПлюс; | б) справочно-правовая система;   |
| 2) MsSQL;           | г) СУБД;                         |
| 3) Adobe Photoshop; | в) графический редактор;         |
| 4) Fine Reader;     | а) система распознавания текста. |

9. Верные ответы: а).

10. Верные ответы: в).

11. Верные ответы: а).

12. Верные ответы: в).

## СЛОВАРЬ ОСНОВНЫХ ПОНЯТИЙ И ТЕРМИНОВ<sup>14</sup>

**ASCII** – американский стандартный код обмена информацией. Широко используется для кодирования в виде байта букв, цифр, знаков операций и других компьютерных символов.

**Bluetooth** – переводится как «голубой зуб» – производственная спецификация беспроводных персональных сетей (PAN). Разговорные названия: синезуб, зуб, блюха.

Bluetooth обеспечивает обмен информацией между такими устройствами как карманные и обычные персональные компьютеры, мобильные телефоны, ноутбуки, принтеры, цифровые фотоаппараты и наушники на надежной, недорогой, повсеместно доступной радиочастоте для ближней связи. Bluetooth позволяет этим устройствам общаться, когда они находятся в радиусе до 10–100 метров друг от друга (дальность очень зависит от преград и помех), даже в разных помещениях.

**File Transfer Protocol (FTP)** – от англ. «протокол передачи файлов» – сетевой протокол, предназначенный для передачи файлов в компьютерных сетях. Протокол FTP позволяет подключаться к серверам FTP, просматривать содержимое каталогов и загружать файлы с сервера или на сервер, кроме того, возможен режим передачи файлов между серверами. FTP является одним из старейших прикладных протоколов, появившимся задолго до HTTP в 1971 г. До начала 90-х гг. на долю FTP приходилось около половины трафика в сети интернет. Этот протокол и сегодня широко используется для распространения программного обеспечения и доступа к удаленным хостам.

**HTML** – от англ. *Hypertext Markup Language* (язык разметки гипертекста) – это стандартный язык разметки документов во Всемирной паутине. Все веб-страницы создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузером и отображается в виде документа, удобном для человека.

HTML является приложением SGML (стандартного обобщенного языка разметки) и соответствует международному стандарту ISO 8879.

---

<sup>14</sup>Более подробно термины, используемые в информатике, можно посмотреть в «Информатика. Энциклопедический систематизированный словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах» (<http://elib.ict.nsc.ru/jspui/handle/ICT/1066>) [2] и в свободной энциклопедии – Википедии (<http://ru.wikipedia.org/wiki/>) [30].

**IP-телефония** (Интернет-Телефония) – технология, которая используется в Интернете для передачи речевых сигналов. В обычном телефонном звонке подключение между собеседниками устанавливается через телефонную станцию исключительно с целью разговора. Голосовые сигналы передаются по определенным телефонным линиям, через выделенное подключение. При запросе же по Интернету, сжатые пакеты данных поступают в Интернет с адресом назначения. Каждый пакет данных проходит собственный путь до адресата, по различным маршрутам. Для адресата, пакеты данных перегруппировываются и декодируются в голосовые сигналы оригинала

**USB-порт** (*Universal Serial Bus*) позволяет подключать устройства с встроенными или съемными модулями памяти без использования дополнительного программного обеспечения. Подключение/отключение устройств от разъема USB можно осуществлять, не выключая основное устройство из сети питания (отсоединение устройства от USB-порта в момент передачи информации может привести к потере данных).

**Wi-Fi** – сокр. от англ. *Wireless Fidelity* – стандарт на оборудование Wireless LAN. Технологию назвали Wireless-Fidelity (дословно «Беспроводная надежность») по аналогии с Hi-Fi. Установка Wireless LAN рекомендуется там, где развертывание кабельной системы невозможно или экономически нецелесообразно. Благодаря функции хендвера пользователи могут перемещаться между точками доступа по территории покрытия сети Wi-Fi. Но при хендвере с одной точки к другой связь ненадолго прерывается. Мобильные устройства (КПК, смартфоны и ноутбуки), оснащенные клиентскими Wi-Fi приемопередающими устройствами, могут подключаться к локальной сети и получать доступ в интернет через так называемые точки доступа или хотспоты.

**XHTML** – от англ. *Extensible Markup Language* (расширяемый язык разметки гипертекста) – язык разметки веб-страниц, по возможностям сопоставимый с HTML, однако является подмножеством XML. Как и HTML, XHTML соответствует спецификации SGML. Вариант XHTML 1.1 одобрен в качестве Рекомендации Консорциума Всемирной паутины (W3C) 31 мая 2001 г.

**XML** – от англ. *eXtensible Markup Language* (расширяемый язык разметки гипертекста) – расширяемый язык разметки; рекомендованный Консорциумом Всемирной паутины язык разметки, фактически представляющий собой свод общих синтаксических правил. XML –



текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки (например, XHTML), иногда называемых словарями. XML является упрощенным подмножеством языка SGML.

Целью создания XML было обеспечение совместимости при передаче структурированных данных между разными системами обработки информации, особенно при передаче таких данных через Интернет. Словари, основанные на XML (например, RDF, RSS, MathML, XHTML, SVG), сами по себе формально описаны, что позволяет программно изменять и проверять документы на основе этих словарей, не зная их семантики, т. е., не зная смыслового значения элементов. Важной особенностью XML также является применение так называемых пространств имен (англ. namespace).

**Автоматизированное рабочее место (АРМ, рабочая станция)** – место оператора, оборудованное всеми средствами, необходимыми для выполнения определенных функций. В системах обработки данных и учреждениях обычно АРМ – это дисплей с клавиатурой, но может использоваться также и принтер, внешние ЗУ и др.

**Автоматизированные обучающие системы (АОС)** – комплексы программно-технических и учебно-методических средств, обеспечивающих активную учебную деятельность: обучение конкретным знаниям, проверку ответов учащихся, возможность подсказки, занимательность изучаемого материала.

**Автоматизированные системы научных исследований (АСНИ)** предназначены для автоматизации научных экспериментов, а также для осуществления моделирования исследуемых объектов, явлений и процессов, изучение которых традиционными средствами затруднено или невозможно.

**Адаптер** – устройство связи компьютера с периферийными устройствами.

**Адрес** – номер конкретного байта оперативной памяти компьютера.

**Алгебра логики (булева алгебра)** – математический аппарат, с помощью которого записывают (кодируют), упрощают, вычисляют и преобразовывают логические высказывания.

**Алгоритм** – понятное и точное предписание (указание) исполнителю совершить определенную последовательность действий для достижения поставленной цели за конечное число шагов.

**Алфавит** – фиксированный для данного языка набор основных символов, т. е. букв алфавита, из которых должен состоять любой текст на этом языке. Никакие другие символы в тексте не допускаются.

**Альфа-тестирование** – от англ. *alpha testing* – использование почти готовой версии продукта (как правило, программного или аппаратного обеспечения) штатными программистами (разработчиками и тестерами) с целью выявления ошибок в его работе для их последующего устранения перед бета-тестированием либо окончательным выходом (выпуском) продукта на рынок, к массовому потребителю.

**Антивирусные программы** – предотвращают заражение компьютерным вирусом и ликвидируют последствия заражения.

**Арифметико-логическое устройство (АЛУ)** – часть процессора, которая производит выполнение операций, предусмотренных данным компьютером.

**Архитектура фон Неймана** – одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд. См. также **Принципы фон-Неймана**.

**Архитектура компьютера** – логическая организация, структура и ресурсы компьютера, которые может использовать программист. Определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера.

**Ассемблер** – см. **Язык ассемблера**

**Аудиоадаптер** (*Sound Blaster*, звуковая плата) – специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

**База данных** – один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

**Байт** (англ. *byte*) – единица измерения количества информации, обычно равная восьми битам (в этом случае может принимать 256 (2<sup>8</sup>) различных значений).

Байт в современных x86 совместимых компьютерах – это минимально адресуемая последовательность фиксированного числа битов.

**Библиотека стандартных подпрограмм** – совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм.

**Бит** – наименьшая единица информации в цифровом компьютере, принимающая значения «0» или «1».

**Блог** – от англ. «*blog*», от *web log*, – «сетевой журнал или дневник событий» – это веб-сайт, основное содержимое которого – регулярно добавляемые записи, изображения или мультимедиа. Для блогов характерны недлинные записи временной значимости. Блоггерами называют людей, ведущих блог. Совокупность всех блогов Сети принято называть блогосферой.

Отличия блога от традиционного дневника обуславливаются средой, т. е. его «сетевостью»: блоги обычно публичны или доступны определенному множеству пользователей Сети. Это определяет и отличия блоговых записей от дневниковых: первые обычно предполагают сторонних читателей, которые могут вступить в публичную полемику с автором (в отзывах к блог-записи или своих блогах). Для блогов характерна возможность публикации отзывов (так называемых комментариев, комментов) посетителями. Она делает блоги средой сетевого общения, имеющей ряд преимуществ перед электронной почтой, группами новостей, веб-форумами и чатами.

**Бета-тестирование** – от англ. *beta testing* – интенсивное использование почти готовой версии продукта (как правило, программного или аппаратного обеспечения) с целью выявления максимального числа ошибок в его работе для их последующего устранения перед окончательным выходом (выпуском) продукта на рынок, к массовому потребителю.

В отличие от альфа-тестирования, проводимого силами штатных разработчиков или тестировщиков, бета-тестирование предполагает привлечение добровольцев из числа обычных будущих пользователей продукта, которым рассылается упомянутая предварительная версия продукта (так называемая бета-версия). Такими добровольцами (их называют бета-тестерами) обычно движет любопытство к новому.

Кроме того, бета-тестирование может использоваться как часть стратегии продвижения продукта на рынок (например, бесплатная раздача бета-версий позволяет привлечь широкое внимание потребителей к окончательной дорогостоящей версии продукта), а также для получения предварительных отзывов о нем от широкого круга будущих пользователей.

**Ввод** – считывание информации с внешнего устройства в память компьютера.

**Вебкамера** (также веб-камера) – цифровая фотокамера, способная в реальном времени фиксировать изображения, предназначенные для дальнейшей передачи по сети Интернет (или в программах типа Instant Messenger, или в любом другом видеоприложении).

**Вентиль** – см. **Логический элемент**.

**Вещественное число** – тип данных, содержащий числа, записанные с десятичной точкой и (или) с десятичным порядком.

**Видеоадаптер** – электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит видеопамять, регистры ввода-вывода и модуль BIOS. Посылает в дисплей сигналы управления яркостью лучей и сигналы развертки изображения.

**Видеокамера** – устройство для получения оптических образов снимаемых объектов на светочувствительном элементе (матрице), приспособленное для записи движущихся изображений. Обычно оснащается микрофоном для параллельной записи звука.

**Винчестер** – см. **Накопитель на жестких магнитных дисках**.

**Вирус компьютерный** – специально написанная небольшая программа, которая может «приписывать» себя к другим программам для выполнения каких-либо вредных действий: порчи файлов, «засорения» оперативной памяти.

**Внешняя память** – в ее состав входят накопители на гибких магнитных дисках (НГМД), накопители на жестких магнитных дисках (винчестерские накопители), накопители на магнитной ленте. Во внешней памяти обычно хранятся архивы программ и данных. Информация, размещенная на внешних носителях (диски, дискеты, кассеты), не зависит от того, включен или выключен компьютер.

**Второе поколение компьютерной техники** – машины, созданные в 1955–1965 гг. Элементная база – дискретные транзисторные логические элементы. Оперативная память на магнитных сердечниках. Высокопроизводительные устройства работы с магнитными лентами, магнитные барабаны и диски. Быстродействие – до сотен тысяч операций в секунду, емкость памяти – до нескольких десятков тысяч слов. Языки высокого уровня, широкий набор библиотечных программ, мониторные системы, управляющие режимом трансляции и исполнения программ.

**Вывод** – результаты работы программы, выдаваемые компьютером пользователю, другому компьютеру или во внешнюю память.

**Выражение** предназначено для выполнения необходимых вычислений. Состоит из констант, переменных, указателей функций, например,  $\sin(x)$ , объединенных знаками операций. Различают выражения арифметические, логические и символьные.

**Гибкий (флоппи) диск** – маленький магнитный диск из гибкой пластмассы для записи и хранения программ и данных.

**Глобальная сеть (ГВС)** – см. **Сеть компьютерная**.

**Графическая плата** известна также как графическая карта, видеокарта, видеоадаптер (англ. *videocard*) – устройство, преобразующее изображение, находящееся в памяти компьютера, в видеосигнал для монитора.

Обычно видеокарта является платой расширения и вставляется в специальный разъем (ISA, VLB, PCI, AGP, PCI-Express) для видеокарт на материнской плате, но бывает и встроенной.

**Графический планшет**, или дигитайзер, диджитайзер – это устройство для ввода рисунков от руки непосредственно в компьютер. Состоит из пера и плоского планшета, чувствительного к нажатию пера. Также может прилагаться специальная мышь.

**Графический редактор** позволяет создавать и редактировать изображения на экране компьютера: рисовать линии, раскрашивать области экрана, создавать надписи различными шрифтами, обрабатывать изображения, полученные с помощью сканеров. Некоторые редакторы обеспечивают возможность получения изображений трехмерных объектов, их сечений и разворотов.

**Графопостроитель (плоттер)** – устройство для вывода из компьютера информации в виде графиков и чертежей на неподвижную или вращающуюся на барабане бумагу.

**Джойстик** – стержень-ручка, отклонение которой от вертикального положения приводит к передвижению курсора в соответствующем направлении по экрану дисплея. Часто применяется в компьютерных играх.

**Диск** – круглая металлическая или пластмассовая пластина, покрытая магнитным материалом, на которую информация наносится в виде концентрических дорожек, разделенных на секторы.

**Дисковод** – устройство, управляющее вращением магнитного диска, чтением и записью данных на нем.

**Дисплей** – устройство визуального отображения информации (в виде текста, таблицы, рисунка, чертежа и др.) на экране электронно-лучевого прибора.

**Драйверы** – программы, расширяющие возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т. д.; с помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся устройств.

**Жесткий диск (винчестер)** – накопитель на жестких магнитных дисках, жесткий диск, хард, HDD или винчестер, (англ. Hard Magnetic Disk Drive, HMDD) – энергонезависимое, перезаписываемое компьютерное запоминающее устройство. Является основным накопителем данных практически во всех современных компьютерах. В отличие от гибкого диска (дискеты), информация в НЖМД записывается на жесткие (алюминиевые или стеклянные) пластины, покрытые слоем ферромагнитного материала, чаще всего двуокиси хрома.

**Звуковая плата** – также называемая как звуковая карта, музыкальная плата (англ. *sound card*) – позволяет работать со звуком на компьютере. В настоящее время звуковые карты бывают встроенными в материнскую плату как отдельные платы расширения и как внешние устройства.

**Идентификатор** – символическое имя переменной, которое идентифицирует ее в программе.

**Инструментальные программные средства** – программы, используемые в ходе разработки, корректировки или развития других программ: редакторы, отладчики, вспомогательные системные программы, графические пакеты и др. По назначению близки системам программирования.

**Интегральная схема** – реализация электронной схемы, выполняющей некоторую функцию, в виде единого полупроводникового кристалла, в котором изготовлены все компоненты, необходимые для осуществления этой функции.

**Интегрированные пакеты программ** – пакеты программ, выполняющие ряд функций, для которых ранее создавались специализированные программы, в частности, текстовые редакторы, электронные таблицы, системы управления базами данных, программы построения графиков и диаграмм.

**Интернет** – гигантская всемирная компьютерная сеть, объединяющая десятки тысяч сетей всего мира. Ее назначение – обеспечить любому желающему постоянный доступ к любой информации. Интернет предлагает практически неограниченные информационные ресурсы, полезные сведения, учебу, развлечения, возможность общения

с компетентными людьми, услуги удаленного доступа, передачи файлов, электронной почты и многое другое. Интернет обеспечивает принципиально новый способ общения людей, не имеющий аналогов в мире.

**Интерпретатор** – разновидность транслятора. Переводит и выполняет программу с языка высокого уровня в машинный код строка за строкой.

**Интерфейс (в вычислительной технике)** – основа взаимодействия всех современных информационных систем. Если интерфейс какого-либо объекта (персонального компьютера, программы, функции) не изменяется (стабилен, стандартизирован), это дает возможность модифицировать сам объект, не перестраивая принципы его взаимодействия с другими объектами.

**Интерфейс пользователя** – совокупность средств, при помощи которых пользователь общается с различными устройствами

**Физический интерфейс** – способ взаимодействия физических устройств. Чаще всего речь идет о компьютерных портах.

**Интерфейсы в программировании** – способ взаимодействия программ.

**Информатизация общества** – организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления организаций, общественных объединений на основе формирования и использования информационных ресурсов.

**Информатика** – дисциплина, изучающая структуру и общие свойства информации, а также закономерности и методы ее создания, хранения, поиска, преобразования, передачи и использования в различных сферах человеческой деятельности. За понятием «информатика» закреплены области, связанные с разработкой, созданием, использованием и материально-техническим обслуживанием систем обработки информации, включая компьютеры и их программное обеспечение, а также организационные, коммерческие, административные и социально-политические аспекты компьютеризации – массового внедрения компьютерной техники во все области жизни людей. Информатика базируется на компьютерной технике.

**Информационная технология** – совокупность методов и устройств, используемых людьми для обработки информации. Охваты-

вает всю вычислительную технику, технику связи и, отчасти, бытовую электронику, телевизионное и радиовещание.

**Информационно-поисковая система (ИПС)** – система, выполняющая функции хранения большого объема информации, быстрого поиска требуемой информации, добавления, удаления и изменения хранимой информации, вывода ее в удобном для человека виде.

**Информация** – сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы (живые организмы, управляющие машины и др.) в процессе жизнедеятельности и работы. Применительно к обработке данных на компьютерах – произвольная последовательность символов, несущих смысловую нагрузку.

**Искусственный интеллект (ИИ)** – дисциплина, изучающая возможность создания программ для решения задач, которые требуют определенных интеллектуальных усилий при выполнении их человеком. Примерами областей использования ИИ являются игры, логический вывод, обучение, понимание естественных языков, формирование планов, понимание речи, доказательство теорем и визуальное восприятие.

**Исполнитель алгоритма** – человек или автомат (в частности, процессор компьютера), умеющий выполнять определенный набор действий. Исполнителя характеризуют среда, элементарные действия, система команд, отказы.

**Источник бесперебойного питания – ИБП** – англ. *UPS-Uninterruptible Power Supply* – автоматическое устройство, позволяющее подключенному оборудованию некоторое время работать от аккумуляторов ИБП, при пропадании электрического тока или при выходе его параметров за допустимые нормы. Кроме того, оно способно корректировать параметры (напряжение, частоту) электропитания. Часто применяется для обеспечения бесперебойной работы компьютеров. Может совмещаться с различными видами генераторов электроэнергии.

**Итерационный цикл** – вид цикла, для которого число повторений операторов тела цикла заранее неизвестно. На каждом шаге вычислений происходит последовательное приближение и проверка условия достижения искомого результата. Выход из цикла осуществляется в случае выполнения заданного условия.

**Кард-ридер** – от англ. *card-rider* – считыватель карт. Устройство для воспроизведения информации с карт памяти различных стан-



дартов. Существуют как внешние (присоединяемые к USB-порту), так и встроенные кард-ридеры.

**Каталог (директория, папка)** – оглавление файлов. Доступен пользователю через командный язык операционной системы. Его можно просматривать, переименовывать зарегистрированные в нем файлы, переносить их содержимое на новое место и удалять. Часто имеет иерархическую структуру.

**Клавиатура компьютера** служит для ввода информации в компьютер и подачи управляющих сигналов. Содержит стандартный набор клавиш печатающей машинки и некоторые дополнительные клавиши – управляющую клавишу, функциональные клавиши, клавиши управления курсором и малую цифровую клавиатуру.

**Клиент (рабочая станция)** – любой компьютер, имеющий доступ к услугам сервера. Клиентом также называют прикладную программу, которая от имени пользователя получает услуги сервера. См. **Сервер**.

**Ключевое слово** – слово языка программирования, имеющее определенный смысл для транслятора. Его нельзя использовать для других целей, например, в качестве имени переменной.

**Команда** – описание элементарной операции, которую должен выполнить компьютер. Обычно содержит код выполняемой операции, указания по определению операндов (или их адресов), указания по размещению получаемого результата. Последовательность команд образует программу.

**Компакт-диск (CD-ROM)** – постоянное ЗУ, выполненное с использованием специальной оптической технологии. В ряду запоминающих устройств занимает место между флоппи- и жестким дисками, являясь одновременно и мобильным и очень емким.

**Компилятор** – разновидность транслятора. Читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

**Компьютер** – программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами. Основу компьютеров образует аппаратура (*HardWare*), построенная, в основном, с использованием электронных и электромеханических элементов и устройств. Принцип действия компьютеров состоит в выполнении программ (*SoftWare*) – заранее заданных, четко определенных последовательностей арифметических, логических и других операций.

**Компьютеризация** – массовое внедрение компьютеров во все области жизни, как необходимое важное условие прогресса и развития, а также последствия, которые будут вызваны этим массовым внедрением компьютеров. Цель компьютеризации – улучшение качества жизни людей за счет увеличения производительности и облегчения условий их труда.

**Контроллер** – устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

**Курсор** – светящийся символ на экране дисплея, указывающий позицию, на которой будет отображаться следующий вводимый с клавиатуры знак.

**Кэш** – см. **Сверхоперативная память**.

**Логический тип** оперирует с данными, допускающими одно из двух возможных значений: «истина» или «ложь» («да» или «нет»). Иногда также называется булевым в честь английского математика XIX века Дж. Буля.

**Логический элемент (вентиль)** – часть электронной логической схемы, выполняющая элементарную логическую функцию.

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

**Локальная сеть (ЛВС)** – см. **Сеть компьютерная**.

**Лэптоп** – компьютер. По размерам близок к обычному портфелю. По основным характеристикам (быстродействие, память) соответствует настольным персональным компьютерам.

**Манипуляторы (джойстик, мышь, трекболл и др.)** – специальные устройства для управления курсором.

**Массив** – последовательность однотипных элементов, число которых фиксировано, которым присвоено одно имя. Компьютерный эквивалент таблицы. Положение элемента в массиве однозначно определяется его индексами.

**Материнская плата** – от англ. *motherboard*, иногда используется сокращение МВ или слово *mainboard* – главная плата – печатная плата, на которой монтируется чипсет и прочие компоненты компьютерной системы.

На материнской плате, кроме чипсета, расположены разъемы для подключения центрального процессора, графической платы, звуковой платы, жестких дисков, оперативной памяти и другие разъемы.

Все основные электронные схемы компьютера и необходимые дополнительные устройства включаются в материнскую плату, или подключаются к ней с помощью слотов расширения.

**Математическая модель** – система математических соотношений – формул, уравнений, неравенств и т. д., отражающих существенные свойства объекта.

**Машинный язык** – совокупность машинных команд компьютера, отличающаяся количеством адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина, и др.

**Меню** – выведенный на экран дисплея список различных вариантов работы компьютера, по которому можно сделать конкретный выбор.

**Микропроцессор** – процессор, выполненный в виде интегральной схемы. Состоит из цепей управления, регистров, сумматоров, счетчиков команд, очень быстрой памяти малого объема.

**Микрокомпьютер** – компьютер, в котором в качестве управляющего и арифметического устройства используется микропроцессор.

**Микрофон** – устройство, позволяющее преобразовывать звук в электрический сигнал и служащее первичным звеном в цепочке звукозаписывающего тракта.

**Модем** – устройство, обеспечивающее преобразование цифровых сигналов компьютера в переменный ток частоты звукового диапазона (модуляцию), а также обратное преобразование (демодуляцию). Используется для соединения компьютера с другими компьютерными системами через телефонную сеть.

**Монитор** – см. **Дисплей**.

**Мультимедиа** – собирательное понятие для различных компьютерных технологий, при которых используется несколько информационных сред, таких, как графика, текст, видео, фотография, движущиеся образы (анимация), звуковые эффекты, высококачественное звуковое сопровождение. *Мультимедиа-компьютер* – это компьютер, снабженный аппаратными и программными средствами, реализующими технологию мультимедиа.

**Мышь** – устройство управления курсором. Имеет вид небольшой коробки, вмещающейся на ладони. Связана с компьютером кабелем. Ее движения трансформируются в перемещения курсора по экрану дисплея.

**Накопитель на жестких магнитных дисках** (винчестерский накопитель) – запоминающее устройство большой емкости, в котором носителями информации являются круглые алюминиевые пластины – платтеры, поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения больших объемов информации.

**Оболочки** – программы, создаваемые для упрощения работы со сложными программными системами, такими, например, как операционная система DOS. Они преобразуют неудобный командный пользовательский интерфейс в дружелюбный графический интерфейс или интерфейс типа «меню». Оболочки предоставляют пользователю удобный доступ к файлам и обширные сервисные услуги.

**Обработка информации** – в информатике – любое преобразование информации из одного вида в другой, производимое по строгим формальным правилам.

**Оперативная память (ОЗУ)** – быстрое запоминающее устройство не очень большого объема, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

**Оператор** – фраза алгоритмического языка, определяющая некоторый законченный этап обработки данных. В состав операторов входят ключевые слова, данные, выражения и др.

**Операционная система** – комплекс взаимосвязанных программ, предназначенных для автоматизации планирования и организации процесса обработки программ, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программ, других вспомогательных операций обслуживания. Важнейшая часть программного обеспечения.

**Описание** – раздел программы, идентифицирующий структуры данных, которыми должна манипулировать программа, и описывающий их типы.

**Оптический привод** – электрическое устройство для считывания и возможно записи информации с оптических носителей (CD-ROM, DVD-ROM).

**Основание системы счисления** – количество различных знаков, используемых для изображения цифр в данной системе.

**Отладка** (англ. *debugging*) – этап компьютерного решения задачи, при котором происходит устранение явных ошибок в програм-

ме. Часто производится с использованием специальных программных средств – отладчиков.

**Отладчик** (англ. *debugger*) – программа, позволяющая исследовать внутреннее поведение разрабатываемой программы. Обеспечивает пошаговое исполнение программы с остановкой после каждой оператора, просмотр текущего значения переменной, нахождение значения любого выражения и др.

**Пакеты прикладных программ (ППП)** – специальным образом организованные программные комплексы, рассчитанные на общее применение в определенной проблемной области и дополненные соответствующей технической документацией.

**Палмтоп (наладонник)** – самый маленький современный персональный компьютер. Умещается на ладони. Магнитные диски в нем заменяет энергонезависимая электронная память. Нет и накопителей на дисках – обмен информацией с обычными компьютерами идет по линиям связи.

**Первое поколение компьютерной техники** – машины, созданные на рубеже 50-х гг. В схемах использовались электронные лампы. Набор команд небольшой, схема арифметико-логического устройства и устройства управления простая, программное обеспечение практически отсутствовало. Быстродействие 10–20 тысяч операций в секунду.

**Переменная** – величина, значение которой может меняться в процессе выполнения программы.

**Персональный компьютер** – микрокомпьютер универсального назначения, рассчитанный на одного пользователя и управляемый одним человеком.

**Подпрограмма** – самостоятельная часть программы, которая создается независимо от других частей и затем вызывается по имени. Когда имя подпрограммы используется в качестве оператора программы, выполняется вся группа операторов, представляющая тело подпрограммы.

**Поколения компьютеров** – условная, нестрогая классификация вычислительных систем по степени развития аппаратных и программных средств, а также способов общения с ними.

**Порты устройств** – электронные схемы, содержащие один или несколько регистров ввода-вывода и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора. Последовательный порт обменивается данными с процессором

побайтно, а с внешними устройствами – побитно. Параллельный порт получает и посылает данные побайтно.

**Постоянная память (ПЗУ)** – используется для хранения данных, не требующих изменения. Содержание памяти специальным образом «зашивается» в ПЗУ при изготовлении. В ПЗУ находятся программа управления работой самого процессора, программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств. Из ПЗУ можно только читать.

**Привод CD** – компакт-диск (CD, Shape CD, CD-ROM, КД ПЗУ) – оптический носитель информации в виде диска с отверстием в центре, информация с которого считывается с помощью лазера. Аббревиатура «CD-ROM» означает *Compact Disc Read Only Memory* что в переводе обозначает компакт-диск с возможностью чтения. CD-ROM-ом часто ошибочно называют CD-привод для чтения компакт-дисков.

**Привод DVD** – (англ. *Digital Versatile Disc* – цифровой многоцелевой диск) – носитель информации в виде диска, внешне схожий с компакт-диском, однако имеющий возможность хранить больший объем информации за счет использования лазера с меньшей длиной волны, чем для обычных компакт дисков.

**Прикладная программа** – любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

**Принтер** – печатающее устройство. Преобразует закодированную информацию, выходящую из процессора, в форму, удобную для чтения на бумаге.

**Принцип открытой архитектуры** –

1. Регламентируются и стандартизируются только описание принципа действия компьютера и его конфигурация (определенная совокупность аппаратных средств и соединений между ними). Таким образом, компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.

2. Компьютер легко расширяется и модернизируется за счет наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым устанавливать конфигурацию своей машины в соответствии со своими личными предпочтениями.

## **Принципы фон-Неймана –**

1. *Принцип программного управления.* Программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

2. *Принцип адресности.* Основная память состоит из перенумерованных ячеек; процессору времени доступна любая ячейка.

3. *Принцип однородности памяти.* Программы и данные хранятся в одной и той же памяти. Поэтому компьютер не различает, что хранится в данной ячейке памяти – число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

**Программное обеспечение (SoftWare)** – совокупность программ, выполняемых компьютером, а также вся область деятельности по проектированию и разработке программ.

**Прокрутка** – имитация программистом за столом выполнения программы на конкретном наборе тестовых данных.

**Протокол коммуникации** – согласованный набор конкретных правил обмена информацией между разными устройствами передачи данных. Имеются протоколы для скорости передачи, форматов данных, контроля ошибок и др.

**Псевдокод** – система обозначений и правил, предназначенная для единообразной записи алгоритмов. Занимает промежуточное место между естественным и формальным языками.

**Регистр** – специальная запоминающая ячейка, выполняющая функции кратковременного хранения числа или команды и выполнения над ними некоторых операций. Отличается от ячейки памяти тем, что может не только хранить двоичный код, но и преобразовывать его.

**Регистр команд** – регистр УУ для хранения кода команды на период времени, необходимый для ее выполнения.

**Рунет** – часть Всемирной паутины, принадлежащая к национальному российскому домену .ru, а также русскоязычные ресурсы, которые могут располагаться в любых доменах (или не иметь домена), а соответствующие серверы могут физически находиться в любой стране мира. К Рунету обычно относят не только WWW-сайты, но и русскоязычные почтовые списки рассылок, IRC-конференции, FTP-серверы, локальные сети разного масштаба и т. п.

В 2001 году слово «Рунет» вошло в орфографический словарь РАН под редакцией В.В. Лопатина – основной словарь государственного языкового портала Грамота.ру; в 2005 – в орфографический словарь Д.Э. Розенталя. В 2004 году была учреждена «Премия Рунета»,

которая через год обрела статус государственной награды под патронажем ФАПМК РФ.

**Сверхоперативная память** – очень быстрое ЗУ малого объема. Используется для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

**Семантика** – система правил истолкования отдельных языковых конструкций. Определяет смысловое значение предложений языка. Устанавливает, какие последовательности действий описываются теми или иными фразами языка и какой алгоритм определен данным текстом на алгоритмическом языке.

**Сенсорный экран** – позволяет осуществлять общение с компьютером путем прикосновения пальцем к определенному месту экрана монитора.

**Сервер** – высокопроизводительный компьютер с большим объемом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования). См. также **Клиент**.

**Сетевая плата** (также известная как сетевая карта, сетевой адаптер, Ethernet-адаптер, NIC (англ. *network interface card*, НВА)) – периферийное устройство, позволяющее компьютеру взаимодействовать с другими устройствами сети.

**Сеть компьютерная** – совокупность трех компонент: 1) сети передачи данных, включающей в себя каналы передачи данных и средства коммутации; 2) компьютеров, взаимосвязанных сетью передачи данных; 3) сетевого программного обеспечения. Пользователи компьютерной сети получают возможность совместно использовать ее программные, технические, информационные и организационные ресурсы.

По степени географического распространения сети делятся на локальные, городские, корпоративные, глобальные и др.

*Локальная сеть (ЛВС)* связывает ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия.

*Глобальная сеть (ГВС)* соединяет компьютеры, удаленные географически на большие расстояния друг от друга. Отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.).



*Городская сеть* обслуживает информационные потребности большого города.

**Синтаксис** – набор правил построения фраз языка, позволяющий определить, какие комбинации символов являются осмысленными предложениями в этом языке.

**Система команд** – совокупность операций, выполняемых некоторым компьютером.

**Система программирования** – система для разработки новых программ на конкретном языке программирования. Предоставляет пользователю мощные и удобные средства разработки программ: транслятор, редактор текстов программ, библиотеки стандартных программ, отладчик и др.

**Система счисления** – способ записи чисел с помощью заданного набора специальных знаков (цифр).

**Система телеконференций** – основанная на использовании компьютерной техники система, позволяющая пользователям, несмотря на их взаимную удаленность в пространстве, а иногда, и во времени, участвовать в совместных мероприятиях, таких, как организация и управление сложными проектами.

**Система управления базами данных (СУБД)** – система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ конечных пользователей.

**Системный блок** – корпус, в котором находятся основные функциональные компоненты персонального компьютера. Корпуса обычно созданы из деталей на основе стали, алюминия и пластика, иногда используют такие материалы как дерево или органическое стекло.

**Системные программы** – программы общего пользования, выполняемые вместе с прикладными и служащие для управления ресурсами компьютера – центральным процессором, памятью, вводом-выводом.

**Системы автоматизированного проектирования (САПР)** – предназначены для выполнения проектных работ с применением компьютерной техники. Широко используются в архитектуре, электронике, механике и др. В качестве входной информации используют технические знания специалистов, которые вводят проектные требования, уточняют результаты, проверяют полученную конструкцию, изменяют ее и т. д. В САПР накапливается информация, поступаю-

щая из библиотек стандартов (данные о типовых элементах конструкций, их размерах, стоимости и др.).

**Системы деловой графики** – позволяют выводить на экран различные виды графиков и диаграмм: гистограммы, круговые и секторные диаграммы и т. д.

**Системы научной и инженерной графики** позволяют в цвете и в заданном масштабе отображать на экране графики двумерных и трехмерных функций, заданных в табличном или аналитическом виде, системы изолиний, в том числе и нанесенные на поверхность объекта, сечения, проекции, карты и др.

**Сканер** – устройство для ввода в компьютер документов: текстов, чертежей, графиков, рисунков, фотографий. Создает оцифрованное изображение документа и помещает его в память компьютера.

**Смартфон** (от англ. *smartphone* – интеллектуальный телефон) – устройство, совмещающее функции мобильного телефона и карманного персонального компьютера (КПК). Отличительной особенностью смартфона от телефона является наличие достаточно развитой операционной системы. Это дает возможность устанавливать дополнительные приложения, расширяющие возможности устройства, которые могут создаваться как фирмой, выпускающей смартфон, так и сторонними разработчиками. КПК, оснащенный функциями GSM-связи называется коммуникатором, смартфон – телефон, оснащенный функциями карманного персонального компьютера.

В настоящее время существует тенденция размывания границ между понятиями «смартфон» и «коммуникатор».

**Сопровождение программ** – работы, связанные с обслуживанием программ в процессе их эксплуатации.

**Стандарт PictBridge** разработан для возможности печати цифровых фотоснимков с фотокамер непосредственно на принтер, без необходимости использования персонального компьютера. Технологическую цепочку печати цветных фотографий таким образом можно не только существенно сократить, но и полностью автоматизировать. Стандарт открытый – все устройства, поддерживающие его, должны работать друг с другом независимо от того, кто является их производителем. До появления PictBridge некоторые производители имели собственные аналогичные стандарты. На сегодняшний день *PictBridge* стал настолько популярен, что его встраивают не только в цифровые фотоаппараты, но и в сотовые телефоны, оборудованные простейшими камерами.

**Стример** – устройство для резервного копирования больших объемов информации. В качестве носителя применяются кассеты с магнитной лентой емкостью 1–2 Гб и больше.

**Структурное программирование** – метод разработки программ, в частности, требующий разбиения программы на небольшие независимые части (модули). Обеспечивает возможность проведения строгого доказательства правильности программ, повышает уверенность в правильности конечной программы.

**Сумматор** – электронная логическая схема, выполняющая суммирование двоичных чисел.

**Суперкомпьютер** – очень мощный компьютер с производительностью свыше 100 мегафлопов (1 мегафлоп – миллион операций с плавающей точкой в секунду). Представляет собой многопроцессорный и (или) многомашинный комплекс, работающий на общую память и общее поле внешних устройств. Архитектура основана на идеях параллелизма и конвейеризации вычислений.

**Схема алгоритма (блок-схема)** – графическое представление алгоритма в виде последовательности блоков, соединенных стрелками.

**Счетчик команд** – регистр устройства управления, содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки команд программы из последовательных ячеек памяти.

**Таблица истинности** – табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

**Тачпад** (англ. *touchpad* – сенсорная площадка, сенсорная панель) – устройство ввода, применяемое вместо мыши, чаще всего в ноутбуках.

**Текстовый редактор** – программа для ввода и изменения текстовых данных (документов, книг, программ, ... ). Обеспечивает редактирование строк текста, контекстный поиск и замену частей текста, автоматическую нумерацию страниц, обработку и нумерацию сносок, выравнивание краев абзаца, проверку правописания слов и подбор синонимов, построение оглавлений, распечатку текста на принтере и др.

**Тест** – некоторая совокупность данных для программы, а также точное описание всех результатов, которые должна выработать про-

грамма на этих данных, в том виде, как эти результаты должны быть выданы программой.

**Тестирование** – этап решения задачи на компьютере, в процессе которого проверяется работоспособность программы, не содержащей явных ошибок.

**Тип данных** – классификация констант, переменных и других элементов данных в зависимости от того, какие величины они представляют: целые, вещественные числа, логические, строковые или символьные значения.

**Топология сети** – способ соединения компьютеров в сеть. Наиболее распространенные виды топологий: линейная, кольцевая, древовидная, звездообразная, ячеистая, полносвязная.

**Транслятор** – программа-переводчик. Преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

**Трекболл** – устройство управления курсором. Небольшая коробка с шариком, встроенным в верхнюю часть ее корпуса. Пользователь рукой вращает шарик и перемещает, соответственно, курсор.

**Третье поколение компьютерной техники** – семейства программно совместимых машин с развитыми операционными системами. Обеспечивают мультипрограммирование. Быстродействие внутри семейства от нескольких десятков тысяч до миллионов операций в секунду. Емкость оперативной памяти – несколько сотен тысяч слов. Элементная база – интегральные схемы.

**Триггер** – электронная схема, широко применяемая в регистрах компьютера для надежного запоминания одного бита информации. Имеет два устойчивых состояния, которые соответствуют двоичной «1» и двоичному «0».

**Тюнер TV** – англ. *TV tuner* – род тюнера, предназначенный для приема телевизионного сигнала в различных форматах вещания (PAL, SECAM, NTSC) с показом на компьютере или просто на отдельном мониторе. Такие тюнеры могут представлять собой как отдельное устройство с радиовходом и аудиовидеовыходами, так и встраиваемую плату. По конструктивному исполнению ТВ-тюнеры бывают внешние (подключаются к компьютеру либо через USB, либо между компьютером и дисплеем через видеокабель) и внутренние (вставляются в слот ISA, или PCI, или PCI-Express).

**Упаковщики (архиваторы)** – программы, позволяющие записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл.

**Устройства ввода-вывода** – класс устройств в типовой архитектуре ЭВМ, предоставляющих компьютеру возможность взаимодействия с внешним миром (с пользователями, а также с другими компьютерами).

**Устройство управления (УУ)** – часть процессора, выполняющая функции управления устройствами компьютера.

**Файл** – место постоянного хранения информации: программ, данных, текстов, закодированных изображений и др. Реализуется как участок памяти на внешних магнитных носителях. Имеет имя, зарегистрированное в каталоге.

**Центральный процессор (ЦП)** – центральное процессорное устройство (ЦПУ) (англ. *central processing unit – CPU*) – процессор машинных инструкций, часть аппаратного обеспечения компьютера или программируемого логического контроллера, отвечающая за выполнение основной доли работ по обработке информации – вычислительный процесс. Современные ЦПУ, выполняемые в виде отдельных микросхем (чипов), реализующих все особенности, присущие данного рода устройствам, называют микропроцессорами. С середины 80-х последние практически вытеснили прочие виды ЦПУ, вследствие чего термин стал все чаще и чаще восприниматься как обыкновенный синоним слова «микропроцессор». Тем не менее, это не так, а центральные процессорные устройства некоторых суперкомпьютеров даже сегодня представляют собой сложные комплексы больших (БИС) и сверхбольших (СБИС) интегральных схем.

**Цикл** – прием в программировании, позволяющий многократно повторять одну и ту же последовательность команд (операторов).

**Цифровой фотоаппарат** – устройство, являющееся разновидностью фотоаппарата, в котором светочувствительный материал заменен светочувствительной матрицей (фотосенсором), состоящей из отдельных пикселей, квантующих изображение по световому параметру. Современные компактные цифровые фотоаппараты многофункциональны, способны записывать кроме фотографий, звук и видео.

**Четвертое поколение компьютерной техники** – теперешнее поколение машин, разработанных после 1970 г. Эти компьютеры проектировались в расчете на эффективное использование современных высокоуровневых языков и упрощение процесса программирования для конечного пользователя. Элементная база – интегральные схемы. Емкость ОЗУ – десятки Мегабайт. Машины этого поколения

представляют собой персональные компьютеры, либо многопроцессорные и (или) многомашинные комплексы, работающие на общую память и общее поле внешних устройств. Быстродействие до нескольких десятков-сотен миллионов операций в секунду.

**Штриховой код (бар-код)** – серия широких и узких линий, в которых зашифрован номер торгового изделия. Имеет большое распространение в организации компьютерного обслуживания торговых предприятий.

**Экспертная система** – комплекс компьютерного программного обеспечения, помогающий человеку принимать обоснованные решения. Использует информацию, полученную заранее от экспертов – людей, которые в какой-либо области являются лучшими специалистами. Хранит знания об определенной предметной области. Обладает комплексом логических средств для выведения новых знаний, выявления закономерностей, обнаружения противоречий и др.

**Электронный офис** – система автоматизации работы учреждения, основанная на использовании компьютерной техники.

**Электронная почта** – система пересылки сообщений между пользователями вычислительных систем, в которой компьютер берет на себя все функции по хранению и пересылке сообщений.

**Электронная таблица** – программа, обрабатывающая таблицы, состоящие из строк и граф, на пересечении которых располагаются клетки. В клетках содержится числовая информация, формулы или текст. Значение в числовой клетке таблицы либо записано, либо рассчитано по формуле. В формуле могут присутствовать обращения к другим клеткам.

**Язык ассемблера** – система обозначений, используемая для представления в удобочитаемой форме программ, записанных в машинном коде. Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая называется *ассемблером* и является, по сути, простейшим транслятором.

**Язык высокого уровня** – язык программирования, более близкий к естественному языку, чем машинный код или язык ассемблера. Каждый оператор в нем соответствует нескольким командам машинного кода или языка ассемблера.

## ЗАКЛЮЧЕНИЕ

Учебное пособие направлено на изучение теоретических основ информатики. Рассмотрены основные понятия информатики: алфавит, слово, информация, сообщение, намерение сообщений и информации, виды и свойства информации, меры количества информации (по Хартли и Шеннону), их свойства и значение, вопросы, связанные с информационными системами и управлением в системе.

В учебном пособии изложены информационные основы, технические и программные средства реализации информационных процессов, компьютерные сети, основы алгоритмизации и программирования. Подробно рассматриваются приложения Microsoft Office: текстовый процессор Word, табличный процессор Excel, система управления базами данных Access, программа подготовки презентаций Power Point.

Уделено особое внимание вопросам алгоритмизации и программирования. Освещение этих вопросов сопровождается большим количеством примеров, наглядно показывающих различные способы программирования вычислительных процессов и характерные особенности в применении тех или иных алгоритмов.

В пособии представлены задания для самостоятельной работы студентов с целью отработки теоретических знаний и формирование практических навыков. Структура учебного пособия позволяет концентрировать внимание студентов на проблемных и перспективных вопросах, последовательно осваивать учебный материал.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Волк, В. К. Информатика: учебное пособие / В. К. Волк. – Москва: Юрайт, 2022. – 207 с. – URL: <https://urait.ru/bcode/496784>.
2. Воройский, Ф. С. Информатика. Энциклопедический систематизированный словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах / Ф. С. Воройский. – Москва: Физматлит, 2006. – 767 с. – URL: <http://elib.ict.nsc.ru/jspui/handle/ICT/1066>
3. Гайдель, А. В. Основы информатики: учебное пособие / А. В. Гайдель. – Самара: Самарский университет, 2019. – 204 с. – URL: <https://e.lanbook.com/book/148609> (дата обращения: 31.05.2022). – Режим доступа: для авториз. пользователей.
4. Гостев, И. М. Операционные системы: учебник и практикум / И. М. Гостев. – Москва: Юрайт, 2022. – 164 с. – URL: <https://urait.ru/bcode/490157>.
5. Гришин, В. А. Теоретические основы информатики. Программное и аппаратное обеспечение: учебное пособие / В. А. Гришин, М. С. Тихов. – Нижний Новгород: ННГУ им. Н. И. Лобачевского, 2019. – 61 с. – URL: <https://e.lanbook.com/book/144952>.
6. Илюшечкин, В. М. Основы использования и проектирования баз данных: учебник / В. М. Илюшечкин. – Москва: Юрайт, 2022. – 213 с. – URL: <https://urait.ru/bcode/488604>.
7. Информатика для экономистов. Практикум: учебное пособие / В. И. Завгородний [и др.]; под редакцией В. И. Завгороднего. – Москва: Юрайт, 2022. – 298 с. – URL: <https://urait.ru/bcode/488830>.
8. Информатика для экономистов: учебник / В. П. Поляков [и др.]; под редакцией В. П. Полякова. – Москва: Юрайт, 2022. – 524 с. – URL: <https://urait.ru/bcode/488884>.
9. Колдовский, Н. Эволюция компьютерной индустрии / Н. Колдовский // URL: [https://overclockers.ru/lab/show/15576/Evoljuciya\\_kompjuternoj\\_industrii](https://overclockers.ru/lab/show/15576/Evoljuciya_kompjuternoj_industrii).
10. Колошкина, И. Е. Компьютерная графика: учебник и практикум для вузов / И. Е. Колошкина, В. А. Селезнев, С. А. Дмитроченко. – Москва: Юрайт, 2022. – 233 с. – URL: <https://urait.ru/bcode/490997>.
11. Кудинов, Ю. И. Основы современной информатики: учебное пособие / Ю. И. Кудинов, Ф. Ф. Пашенко. – Санкт-Петербург: Лань, 2022. – 256 с. – URL: <https://e.lanbook.com/book/213647>.



12. Малиновский, Б. Н. Из мировой истории цифровой вычислительной техники / Б. Н. Малиновский // URL: <https://www.computer-museum.ru/frgnhist/malinovs.htm>
13. Малов, А. В. Концепции современного программирования: учебное пособие / А. В. Малов, С. В. Родионов. – Москва: Юрайт, 2022. – 96 с. – URL: <https://urait.ru/bcode/485436>.
14. Медникова, О. В. Теоретические основы информатики: учебно-методическое пособие / О. В. Медникова. – Москва, 2019. – 56 с. – URL: <https://e.lanbook.com/book/175763>.
15. Новожилов, О. П. Архитектура ЭВМ и систем в 2 ч. Часть 1: учебное пособие / О. П. Новожилов. – Москва: Юрайт, 2022. – 276 с. – URL: <https://urait.ru/bcode/494314>.
16. Новожилов, О. П. Архитектура ЭВМ и систем в 2 ч. Часть 2: учебное пособие / О. П. Новожилов. – Москва: Юрайт, 2022. – 246 с. – URL: <https://urait.ru/bcode/494315>.
17. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – Санкт-Петербург: Питер, 2016. – 996 с.
18. Попова, О. В. Информатика: учебное пособие / О. В. Попова. – Красноярск: Красноярский институт экономики Санкт-Петербургской академии управления и экономики (НОУ ВПО), 2007. – 186 с. – URL: <http://www.klyaksa.net/htm/kopilka/uchp/index.htm>
19. Процедурно-ориентированное программирование // URL: [https://www.intuit.ru/department/pl/javapl/2/javapl\\_2.html](https://www.intuit.ru/department/pl/javapl/2/javapl_2.html).
20. Симонович, С. В. Информатика. Базовый курс: учебник / С. В. Симонович. – Санкт-Петербург: Питер, 2011. – 640 с.
21. Советов, Б. Я. Базы данных: учебник / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – Москва: Юрайт, 2022. – 420 с. – URL: <https://urait.ru/bcode/488866>.
22. Софронова, Н. В. Теория и методика обучения информатике: учебное пособие / Н. В. Софронова, А. А. Бельчусов. – Москва: Юрайт, 2022. – 401 с. – URL: <https://urait.ru/bcode/492641>.
23. Толстобров, А. П. Архитектура ЭВМ: учебное пособие / А. П. Толстобров. – Москва: Юрайт, 2022. – 154 с. – URL : <https://urait.ru/bcode/496167>.
24. Торадзе, Д. Л. Информатика: учебное пособие / Д. Л. Торадзе. – Москва: Юрайт, 2022. – 158 с. – URL: <https://urait.ru/bcode/496823>.

25. Трофимов, В. В. Алгоритмизация и программирование: учебник / В. В. Трофимов, Т. А. Павловская. – Москва: Юрайт, 2022. – 137 с. – URL: <https://urait.ru/bcode/491215>.
26. Трофимов, В. В. Информатика в 2 т. Том 1: учебник / В. В. Трофимов, М. И. Барабанова. – Москва: Юрайт, 2022. – 553 с. – URL: <https://urait.ru/bcode/470744>.
27. Хеннесси Джон Л., Паттерсон Дэвид А. Компьютерная архитектура. Количественный подход / Джон Л. Хеннесси, Дэвид А. Паттерсон. – Москва: ТЕХНОСФЕРА, 2016. – 936 с.
28. Шауцукова, Л. З. Информатика. Теория. Практика / Л. З. Шауцукова // URL: <http://www.kbsu.ru/~book>.
29. Экономическая информатика: учебник и практикум / Ю. Д. Романова [и др.]. – Москва: Юрайт, 2022. – 495 с. – URL: <https://urait.ru/bcode/508139>.
30. Свободная энциклопедия – Википедия // URL: <http://ru.wikipedia.org/wiki/72>.
31. [https://www.cnews.ru/news/top/2021-11-16\\_rossijskie\\_superkompyutery](https://www.cnews.ru/news/top/2021-11-16_rossijskie_superkompyutery).
32. <http://www.intuit.ru/department/hardware/archsys/popup.lit.html#1>.
33. <https://www.kommersant.ru/doc/4147760>.
34. <https://www.ixbt.com/news/2021/11/01/nazvany-samye-populyarnye-brauzery-v-mire--v-rossii-kartina-silno-otlichaetsja-ot-obshemirovoj.html>.
35. <https://remontcompa.ru/zhelezo-i-periferiya/2604-vyshli-processory-intel-core-12-go-pokolenija-alder-lake.html>.
36. <https://www.top500.org/lists/top500/list/2021/11>.
37. <https://www.dxdigitals.info/2014/02/vibor-processora-vibirayem-samiy-luchshiy-processor.html>.

# **ВВЕДЕНИЕ В ИНФОРМАТИКУ (БАЗОВЫЙ КУРС)**

*Учебное пособие*

*Титовская Наталья Викторовна  
Титовский Сергей Николаевич  
Болдарук Ирина Ивановна  
Амбросенко Николай Дмитриевич*

*Электронное издание*

Редактор М. М. Ионина

Подписано в свет 01.09.2022. Регистрационный номер 105  
Редакционно-издательский центр Красноярского государственного аграрного университета  
660017, Красноярск, ул. Ленина, 117  
e-mail: rio@kgau.ru