

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ

«БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

**Кафедра машиностроения и транспорта**

УТВЕРЖДАЮ:

Проректор по учебной работе

\_\_\_\_\_ Е.И.Луковникова

« \_\_\_\_\_ » декабря 2018 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**ИНФОРМАТИКА**

**Б1. Б.13**

**НАПРАВЛЕНИЕ ПОДГОТОВКИ**

**23.03.03 Эксплуатация транспортно-технологических  
машин и комплексов**

**ПРОФИЛЬ ПОДГОТОВКИ**

**Автомобили и автомобильное хозяйство**

Программа академического бакалавриата

Квалификация (степень) выпускника: бакалавр

<b>1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ .....</b>	<b>3</b>
<b>2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ .....</b>	<b>4</b>
<b>3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ .....</b>	<b>4</b>
3.1 Распределение объёма дисциплины по формам обучения.....	4
3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости.....	4
<b>4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ .....</b>	<b>5</b>
4.1 Распределение разделов дисциплины по видам учебных занятий .....	5
4.2 Содержание дисциплины, структурированное по разделам и темам .....	6
4.3 Лабораторные работы.....	45
4.4 Практические занятия.....	45
4.5 Контрольные мероприятия: контрольная работа.....	46
<b>5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>47</b>
<b>6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ</b>	<b>48</b>
<b>7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>48</b>
<b>8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>49</b>
<b>9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....</b>	<b>49</b>
9.1. Методические указания для обучающихся по выполнению лабораторных работ..	49
9.2. Методические указания по выполнению курсовой работы .....	108
<b>10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ .....</b>	<b>109</b>
<b>11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ .....</b>	<b>109</b>
<b>Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....</b>	<b>110</b>
<b>Приложение 2. Аннотация рабочей программы дисциплины .....</b>	<b>124</b>
<b>Приложение 3. Протокол о дополнениях и изменениях в рабочей программе .....</b>	<b>125</b>

# 1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

## Вид деятельности выпускника

Дисциплина охватывает круг вопросов, относящихся к производственно-технологическому и экспериментально-исследовательскому виду профессиональной деятельности выпускника в соответствии с компетенциями и видами деятельности, указанными в учебном плане.

## Цель дисциплины

Целью изучения дисциплины является ознакомление с методами, способами получения, хранения и обработки информации; получение практических навыков в использовании современного прикладного программного обеспечения и возможностей глобальных компьютерных сетей.

## Задачи дисциплины:

- использования информационных технологий при проектировании и разработке в составе коллектива исполнителей новых видов транспорта и транспортного оборудования, а также транспортных изделий;
- эффективного использования материалов, оборудования, соответствующих алгоритмов и программ расчетов параметров технологических процессов;
- информационного поиска и анализа информации по объектам исследований;
- участия в составе коллектива исполнителей в обосновании и применении новых информационных технологий.

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
1	2	3
ОПК-1	способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- методы и процессы сбора, передачи, обработки и накопления информации;</li> <li>- технические и программные средства реализации информационных процессов;</li> <li>- модели решения функциональных и вычислительных задач;</li> <li>- алгоритмизацию и программирование;</li> <li>- языки программирования;</li> <li>- базы данных;</li> <li>- программное обеспечение и технологию программирования,</li> <li>- компьютерную графику;</li> <li>- локальные сети и их использование при решении прикладных задач обработки данных;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>-использовать возможности вычислительной техники и программного обеспечения в отрасли</li> </ul> <p><b>владеть:</b></p> <ul style="list-style-type: none"> <li>-пользовательскими вычислительными системами и системами программирования с учетом основных требований информационной безопасности.</li> </ul>

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.Б.13 «Информатика» относится к базовой части.

Дисциплина «Информатика» базируется на знаниях, полученных при изучении учебных дисциплин основных общеобразовательных программ.

Основываясь на изучении перечисленных дисциплин, дисциплина «Информатика» представляет основу для изучения дисциплины «Вычислительная техника и сети в отрасли».

Такое системное междисциплинарное изучение направлено на достижение требуемого ФГОС уровня подготовки по квалификации бакалавр.

## 3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ

### 3.1. Распределение объема дисциплины по формам обучения

Форма обучения	Курс	Семестр	Трудоемкость дисциплины в часах						Курсовая работа	Вид промежуточной аттестации
			Всего часов (с экз.)	Аудиторных часов	Лекции	Лабораторные работы	Семинары Практические занятия	Самостоятельная работа		
1	2	3	4	5	6	7	8	9	10	11
Очная	-	-	-	-	-	-	-	-	-	-
Заочная	1	-	144	16	8	8	-	119	кр	экзамен
Заочная (ускоренное обучение)	-	-	-	-	-	-	-	-	-	-
Очно-заочная	-	-	-	-	-	-	-	-	-	-

### 3.2. Распределение объема дисциплины по видам учебных занятий и трудоемкости

Вид учебных занятий	Трудоемкость (час.)	в т.ч. в интерактивной, активной, инновационной формах, (час.)	Распределение по курсам, час
			1
<b>I. Контактная работа обучающихся с преподавателем (всего)</b>	16	4	34
Лекции (Лк)	8	-	17
Лабораторные работы (ЛР)	8	4	17
Контрольная работа	+	-	+
Групповые (индивидуальные) консультации*	+	-	+
<b>II. Самостоятельная работа обучающихся (СР)</b>	119	-	119
Подготовка к лабораторным работам	8	-	8
Подготовка к экзамену в течение семестра	93	-	93
Подготовка к зачету	-	-	-
Выполнение контрольной работы	18	-	18
<b>III. Промежуточная аттестация экзамен</b>	9		9
Общая трудоемкость дисциплины ..... час.	144	-	144
зач. ед.	4	-	4

## 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1. Распределение разделов дисциплины по видам учебных занятий

- для заочной формы обучения:

№ раздела и темы	Наименование раздела и тема дисциплины	Трудоемкость, (час.)	Виды учебных занятий, включая самостоятельную работу обучающихся, и трудоемкость; (час.)		
			учебные занятия		самостоятельная работа обучающихся*
			лекции	лабораторные работы	
1	2	3	4	5	6
<b>1.</b>	<b>Общая характеристика процессов сбора, передачи, обработки и накопления информации.</b>	<b>16</b>	<b>2</b>	<b>-</b>	<b>14</b>
1.1.	Основные понятия информатики	5,5	0,5	-	5
1.2.	Понятие информации. Виды информации.	2,5	0,5	-	2
1.3.	Единицы измерения и носители информации.	2,5	0,5	-	2
1.4.	Понятие файла.	3	0,5	-	2
<b>2.</b>	<b>Локальные и глобальные сети ЭВМ.</b>	<b>18</b>	<b>2</b>	<b>-</b>	<b>16</b>
2.1.	Сети ЭВМ	2,5	0,5	-	2
2.2.	Мировая компьютерная сеть	3	1	-	2
2.3.	Вредоносное программное обеспечение	2,5	0,5	-	2
<b>3.</b>	<b>Технические средства реализации информационных процессов</b>	<b>18</b>	<b>2</b>	<b>-</b>	<b>16</b>
3.1.	Основы устройства ЭВМ	2,5	0,5	-	2
3.2.	ПЭВМ, ее устройство и основные технические характеристики	3	1	-	2
3.3.	Рабочие станции и серверы.	2,5	0,5	-	2
<b>4.</b>	<b>Программные средства реализации информационных процессов</b>	<b>53</b>	<b>1</b>	<b>4</b>	<b>48</b>
4.1.	Операционные системы.	3,5	0,5	1	2
4.2.	Сервисное программное обеспечение.	6,5	0,5	2	4
4.3.	Офисное программное обеспечение.	4,5	-	0,5	4
4.4.	Программное обеспечение для работы с компьютерной графикой.	4,5	-	0,5	4
4.5.	Системы управления базами данных	4	-	-	4
<b>5.</b>	<b>Алгоритмизация и программирование</b>	<b>30</b>	<b>1</b>	<b>4</b>	<b>25</b>
5.1.	Понятие алгоритма.	3,5	0,5	1	2
5.2.	Языки программирования.	3,5	0,5	1	2
5.3.	Стили программирования.	3	-	1	2
5.4.	Понятие об объектноориентированном программировании.	5	-	1	4
	<b>ИТОГО</b>	<b>135</b>	<b>8</b>	<b>8</b>	<b>119</b>

## 4.2. Содержание дисциплины, структурированное по разделам и темам

### Раздел 1. Общая характеристика процессов сбора, передачи, обработки и накопления информации

#### Тема 1.1. Основные понятия информатики

На сегодняшний день информатика стала одной из самых популярных научных дисциплин. Вопросам этой науки посвящено много книг, журналов, различных публикаций. Однако, не всегда люди, интересующиеся проблемами информатики, могут достаточно четко определить круг вопросов, которые охватывает информатика как наука.

Часто бытует мнение, что эта дисциплина включает в себя лишь задачи программирования, обработки данных или учение о вычислительных машинах. Да, информатика этим занимается, но частично. На самом деле эта наука включает в себя множество математических, инженерных и даже философских аспектов, через которые она становится фундаментальной наукой, занимающейся схематичным, "формализованным" представлением информации, вопросами ее обработки, а также различными средствами, с помощью которых можно производить необходимую обработку информации. Это включает в себя вопросы анализа и моделирования взаимосвязей и структур в самых различных областях применения. При этом возникает необходимость в разработке способов решения задач информационной обработки на вычислительных машинах, а также в разработке, организации и эксплуатации самих вычислительных машин и систем.

Формирование моделей информатики нацелено на представление определенных структур, взаимодействий и процессов в какой-либо области применения с помощью формальных средств - таких как, структуры данных, языки программирования или логические формулы.

Задача информатики состоит в том, чтобы исследовать свойства формальных моделей и развивать их дальше, и не в последнюю очередь - устанавливать связи между формальными моделями и реальным миром в данной предметной области в смысле постановки задачи.

Таким образом, на основании вышеизложенных фактов можно дать следующее определение информатики:

Информатика - это наука и техника, связанные с машинной обработкой, хранением и передачей информации.

Следует заметить, что в других источниках информации можно найти иные определения информатики. Они, наверняка, тоже верны. Почему возникает такая ситуация?

Во-первых, дело в том, что информатика - наука многогранная. Как говорилось выше, в ней рассматриваются различные аспекты - от математических до философских. Поэтому в определении информатики автор может сделать упор на тот аспект, которому уделяется больше внимания или который он считает более важным.

Во-вторых, следует отметить, что информатика - наука развивающаяся. Следовательно, как любое развивающееся явление, информатика постоянно претерпевает изменения. А это естественным образом влечет за собой изменения понятий, терминов, определений.

#### Предметная область информатики

Отметим, что предметная область любой науки определяется (как это принято в философии и науковедении) не только областью действительности, изучаемой этой наукой, но и методологией, методами исследования этой области. Характеризуя методологию информатики, приведем позицию по этому вопросу Института проблем информатики РАН, сформулированную в работе "Развитие определений "информатика" и "информационные технологии", вышедшей под редакцией чл. - корр. РАН И.А. Мизина. В ней отмечается: "Важнейшим методологическим принципом информатики является изучение объектов и явлений окружающего мира с точки зрения процессов сбора, обработки и выдачи информации о них, а также определенного сходства этих процессов при их реализации в искусственных и естественных (в том числе биологических и социальных) системах".

Из всего вышесказанного вытекает, что общеобразовательный компонент понятия информации раскрывается, по крайней мере, через три основных аспекта:

философских и методологических вопросов, связанных с соотношением понятий: "знание" - "информация";

комплекса научных дисциплин, занимающейся изучением самоуправляемых систем различной природы; изучение процессов сбора, хранения и переработки информации в системах различной природы, в том числе и с помощью компьютера.

Из этих трех основных компонент складывается образовательная область "Информатика".

#### Структура и закономерности протекания информационных процессов

Процесс описывает возможное функционирование распределенной системы. Следовательно, информационный процесс описывает возможное функционирование распределенной информационной системы. Информационные системы - область науки и техники, которая включает совокупность средств, способов и методов человеческой деятельности, направленных на создание и применение систем сбора, хранения, передачи и обработки информации. Процесс в общем случае состоит из множества событий, которые соответствуют выполнению действий и возникают в определенном причинном следственном порядке. Процесс служит для математического моделирования любого дискретного действия распределенной системы и складывается из отдельных действий между которыми существуют причинно-следственные отношения. При каждом воплощении дей-

ствия говорят о событии. Действие распределенной системы состоит из множества событий, причем каждому событию предписано действие.

В случае сложных конечных или бесконечных процессов рассмотрение неструктурированного множества всех событий и связанных с ними действий становится трудным. Часто процесс складывается из ряда частичных процессов (подпроцессов). Такого рода процесс можно анализировать путем разложения на подпроцессы и их анализа.

Последовательные процессы являются частным случаем параллельных процессов. Для представления последовательных процессов предлагается более простая форма представления в виде структуры действий. Для представления последовательных процессов можно полностью отказаться от явного задания множества событий, а вместо них использовать конечную или бесконечную последовательность действий.

Каждому последовательному процессу можно однозначно сопоставить поток действий – трассу (spur, англ. trace) процесса. Трассы являются более простыми моделями для последовательного хода работы системы, чем структуры действий.

Для моделирования распределенных систем, как правило, используются не процессы с их явным представлением параллельности (англ. true concurrency – истинный параллелизм), а технически более простые для применения множества потоков действий. В этом случае говорят об искусственной линейаризации (англ. interleaving).

Абстрактное описание динамического поведения распределенной системы может быть дано путем задания множества процессов. Существует ряд способов для описания систем и их ходов работ:

- сети Петри, графический метод описания;
- агенты, формальный язык описания;
- формулы логики предикатов для описания хода работы.

Для описания процессов применяются следующие термины:

распределение означает пространственное расположение (или разделение) отдельных компонент процесса;

параллелизм относится к временным отношениям между действиями компонентов процесса, которые могут протекать одновременно (параллельно);

интерактивность, реакция, коммуникация, координация, синхронизация касаются причинно-следственных отношений между пространственно разделенными и выполняемыми наряду друг с другом действиями (с помощью них осуществляется обмен сообщениями и сигналами);

недетерминированность возникает при моделировании распределенных процессов из-за целенаправленного опускания из рассмотрения определенных деталей информации, например, временных взаимозависимостей в функционировании процесса, которые являются существенными при принятии решений в ходе работы процесса.

Процесс в общем случае состоит из множества событий, которые соответствуют выполнению действий и возникают в определенном причинно-следственном (временном) порядке.

## **Раздел 1. Общая характеристика процессов сбора, передачи, обработки и накопления информации**

### **Тема 1.2. Понятие информации**

Предметом изучения информатики является одна из исходных категорий мироздания – информация. К исходным категориям мироздания относятся так же материя и энергия. Эти категории взаимосвязаны между собой. Усмотреть такие связи можно как в природных явлениях, так и в процессах и явлениях, порожденных человеком.

К основным понятиям информатики относятся также структура, модель, алгоритм. В любой науке, в том числе информатике, основные понятия определить достаточно сложно. Нельзя однозначно определить точку в геометрии, материю в физике, информацию в информатике.

Понятие «информация» является одним из основных во многих других науках: в журналистике и философии, в экономике и физике, в математике и психологии.

Если попытаться дать определение, следует учесть, что информация:

- существует в неживой природе;
- существует в биологических системах;
- может не всегда быть выражена словами;
- возникает в процессе общения;
- хранится, обрабатывается, передается и т.д.

Понятие информации – одно из самых фундаментальных в современной науке. Наряду с такими понятиями, как вещество, энергия, пространство и время, оно составляет основу современной научной картины мира.

В зависимости от области знания существуют различные подходы к определению понятия информации.

Информация – от латинского informatio – сведения, разъяснения, изложение.

В быту под информацией понимают сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специальными устройствами.

В технике под информацией понимают сообщения, передаваемые в форме знаков или сигналов.

В теории информации – сведения, которые снимают полностью или уменьшают существующую до их получения неопределенность. По определению Клод Элвуд Шеннона, информация – это снятая неопределенность.

В кибернетике по определению Норберта Виннера, понимают ту часть знаний, которая используется для ориентирования, активного действия, управления, т.е. в целях сохранения, совершенствования, развития системы.

В семантической теории (смысл сообщения) понимают сведения, обладающие новизной.

В документалистике - все то, что так или иначе зафиксировано в знаковой форме в виде документов.

В информатике термин информация является базовым, фундаментальным. Определить его исчерпывающим образом через какие-то более простые понятия невозможно. Каждый вариант определения обладает некоторой неполнотой.

В узком смысле слова, под информацией понимают факты, сведения, сообщения.

В более широком смысле информация – это объективное свойство реального мира, отражающее взаимодействие различных объектов.

### **Виды информации**

У человека пять органов чувств, с их помощью человек получает информацию о внешнем мире: зрение, слух, обоняние, вкус, осязание.

Практически около 90% информации человек получает при помощи органов зрения (визуальный), примерно 9% — при помощи органов слуха (аудиальный) и только 1% при помощи остальных органов чувств (обоняния, вкуса, осязания).

Рассмотрим только те виды информации, которые «понимают» технические устройства (в частности, компьютер).

Текстовая; Числовая; Графическая; Звуковая; Мультимедийная (комбинированная).

Виды информации по общественному значению:

личная (знания, умения, навыки, интуиция);

массовая (общественная, обыденная, эстетическая);

специальная (научная, производственная, техническая, управленческая).

Предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств, называются информационными объектами.

Следует различать термины информация и данные.

Данные – это информация, представленная в виде, позволяющем автоматизировать ее сбор, хранение и дальнейшую обработку человеком или информационным средством.

### **Подходы к оценке количества информации**

Понятие информации предполагает наличие двух объектов – источника информации и потребителя. Важно, чтобы информация для потребителя имела смысл. Потребитель информации может ее оценивать в зависимости от того, где и для какой конкретной задачи информация используется. Поэтому выделяют такие аспекты информации:

Прагматический аспект связан с возможностью достижения поставленной цели с использованием получаемой информации. Если информация была эффективной, то поведение потребителя меняется в желаемом направлении, т.е. информация имеет прагматическое содержание. Этот аспект характеризует поведенческую сторону проблемы: рассматриваются отношения между сообщением и получателем.

Семантический аспект информации позволяет оценить смысл передаваемой информации и определяется семантическими связями между словами или другими смысловыми элементами языка. На этом уровне анализируются отношения между знаками и обозначаемыми ими предметами, действиями, качествами, т.е. отношение к источнику информации.

Синтаксический аспект информации связан со способом ее представления. В зависимости от реального процесса, в котором участвует информация (осуществляется ее сбор, передача, преобразование, отражение, представление, ввод или вывод), она представляется в виде специальных знаков, символов. На этом уровне рассматриваются внутренние свойства сообщений, т.е. отношения между знаками, отражающие структуру данной знаковой системы.

Для каждого из аспектов рассмотрения информации существуют свои подходы к измерению количества информации и свои меры информации.

Количественная оценка информации этого уровня не связана с содержательной стороной информации, а оперирует с обезличенной информацией, не выражающей смыслового отношения к объекту.

Для измерения информации на синтаксическом уровне вводятся два параметра: объем информации (данных) –  $V_d$  (объемный подход) и количество информации  $I$  (энтропийный подход).

Объем информации (данных) –  $V_d$  (объемный подход). При реализации информационных процессов информация передается в виде сообщения, представляющего собой совокупность символов какого-либо алфавита. При этом каждый новый символ в сообщении увеличивает количество информации, представленной последовательностью символов данного алфавита. Если количество информации, содержащейся в сообщении из одного символа, принять за единицу, то объем информации (данных) –  $V_d$  в любом другом сообщении будет равен количеству символов (разрядов) в этом сообщении. Одна и та же информация может быть представлена многими разными способами (с использованием разных алфавитов), поэтому и единица измерения информации соответственно будет меняться.

В десятичной системе счисления один разряд имеет вес, равный 10, и соответственно единицей измерения информации будет десятичный разряд – дит. В этом случае сообщение в виде  $n$ -разрядного числа имеет объем информации (данных) –  $V_d = n$  дит. Например, шестиразрядное число 436791 имеет объем данных  $V_d = 6$  дит.

В двоичной системе счисления один разряд имеет вес, равный 2, и соответственно единицей измерения информации будет двоичный разряд – бит (bit – binary digit). В этом случае сообщение в виде n-разрядного числа имеет объем информации (данных) –  $V_d = n$  бит. Например, восьмиразрядное число 10110011 имеет объем данных  $V_d = 8$  бит.

В современной вычислительной технике наряду с минимальной единицей «бит» широко используется укрупненная единица информации байт, равная 8 бит. При работе с большими объемами информации для подсчета ее количества применяют более крупные единицы измерения, которые в системе измерения двоичной информации получаются путем умножения основной единицы на 210, 220 и т.д.

1 Килобайт:  $1 \text{ Кб} = 2^{10} \text{ б} = 1024 \text{ байт}$ ;

Мегабайт:  $1 \text{ Мб} = 2^{10} \text{ Кб} = 1024 \text{ Кб} = 1\,048\,576 \text{ байт}$ ;

Гигабайт:  $1 \text{ Гб} = 2^{10} \text{ Мб} = 1024 \text{ Мб} = 1\,073\,741\,824 \text{ байт}$ ;

Терабайт:  $1 \text{ Тб} = 2^{10} \text{ Гб} = 1024 \text{ Гб} = 1\,099\,511\,627\,776 \text{ байт}$ .

### **Основы и методы защиты информации**

Широкое использование компьютерных и сетевых технологий для обработки и передачи информации поставило перед информатикой вопрос защиты информации от возможного доступа к ней посторонних лиц, её искажения или утраты. На всех этапах работы с информацией, последняя должна быть надежно защищена. Информационная безопасность включает в себя целый комплекс мер. Выделим те, которые являются актуальными для рядового пользователя сегодня с позиции обеспечения сохранности информации.

Основную угрозу представляет Internet, в котором совместно с другой информацией ведёт активную жизнь вредоносное ПО, называемое вирусами. Компьютерный вирус — это программный код или блок программного кода, встроенный в другую программу (документ или в определенные области носителя данных) и предназначенный для выполнения несанкционированных действий на несущем компьютере. При запуске программы, несущей вирус, происходит запуск имплантированного в нее вирусного кода. Работа этого кода вызывает скрытые от пользователя изменения в файловой системе жестких дисков и/или в содержании других программ. Так, например, вирусный код может воспроизводить себя в теле других программ — этот процесс называется размножением. Из наиболее серьезных последствий, связанных с вирусами, является возможность потери либо порчи части данных на диске, в редких случаях порча аппаратного обеспечения. Поэтому на рабочем компьютере должен быть установлен антивирус. При этом следует иметь в виду, что всегда существует вероятность заражения компьютера в связи с неизбежной задержкой обновления антивирусных баз соответствующего ПО, каким бы «продвинутым» оно не являлось.

При работе в сети на компьютере необходимо наличие файервола (firewall), который является интеллектуальным барьером между пользователем и сетью. Он позволяет блокировать самостоятельный выход программ в Интернет и успешно отражать атаки из сети.

Более эффективным средством является резервное копирование важной информации, которое позволяет сохранить информацию не только при её искажении, но и при полной потере, например, при выходе из строя устройств хранения данных в компьютере. Резервное копирование лучше всего выполнять на одноразовые записываемые носители. При наличии необходимости периодически сохранять изменяющиеся данные (например, информация, которая используется при работе предприятия) резервирование рационально делать на другие жесткие диски, стоящие на специально предназначенном для этих целей компьютере.

Средством защиты информации от несанкционированного доступа других людей является шифрование — то есть кодирование информации соответствующими методами без возможности доступа к ней при отсутствии соответствующего ключа. Различают симметричное и несимметричное шифрование. При симметричном шифровании происходит с помощью одного и того же ключа. Основным недостатком симметричного процесса заключается в том, что, прежде чем начать обмен информацией, надо выполнить передачу ключа, а для этого нужна защищенная связь, то есть проблема повторяется, хотя и на другом уровне.

Поэтому в настоящее время в Интернете используют несимметричные криптографические системы, основанные на использовании не одного, а двух ключей. Происходит это следующим образом. Компания для работы с клиентами создает два ключа: один открытый (public — публичный), а другой закрытый (private — личный). На самом деле это как бы две «половинки» одного целого ключа, связанные друг с другом. Количество комбинаций, которое надо проверить при реконструкции закрытого ключа, не столь велико, как количество возможных шахматных партий, однако защиту информации принято считать достаточной, если затраты на ее преодоление превышают ожидаемую ценность самой информации. В этом состоит принцип достаточности защиты, которым руководствуются при использовании несимметричных средств шифрования данных.

Следует иметь в виду, что эффективность защиты информации тем выше, чем шире комплекс мер, принятых для обеспечения сохранности информации.

## **Раздел 1. Общая характеристика процессов сбора, передачи, обработки и накопления информации**

### **Тема 1.3. Единицы измерения и носители информации**

Носитель информации – среда для записи и хранения информации.

Недолговременные носители информации передаются с помощью различных физических устройств – устройств связи. (телефон, телеграф, радио, компьютеры и телеком. сети и т.д.).

Устройство связи состоит из источника (передатчика) информации приемника (получателя).

Между ними действует канал передачи информации – канал связи.

Канал связи – совокупность технических устройств, обеспечивающих передачу сигнала от источника к приемнику.

Кодирующее устройство – устройство, предназначенное для преобразования исходного сообщения источника информации к виду, удобному для передачи.

Декодирующее устройство – устройство для преобразования закодированного сообщения в исходное

Информация - это сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специализированным устройством, например ЭВМ, для обеспечения целенаправленной деятельности.

Информация может быть по своей физической природе: числовой, текстовой, графической, звуковой, видео и др. Она также может быть постоянной (неменяющейся), переменной, случайной, вероятностной. Наибольший интерес представляет переменная информация, так как она позволяет выявлять причинно-следственные связи в процессах и явлениях. Существуют различные способы оценки количества информации. Классическим является подход, использующий формулу К. Шеннона. Применительно к двоичной системе она имеет вид:

$$H = \log_2 N,$$

где  $H$  – количество информации, несущей представление о состоянии, в котором находится объект;  $N$  – количество равновероятных альтернативных состояний объекта.

Любая информация, обрабатываемая в ЭВМ, должна быть представлена двоичными цифрами  $\{0,1\}$ , т.е. должна быть закодирована комбинацией этих цифр. Различные виды информации (числа, тексты, графика, звук) имеют свои правила кодирования. Коды отдельных значений, относящиеся к различным видам информации, могут совпадать. Поэтому расшифровка закодированных данных осуществляется по контексту при выполнении команд программы.

### Кодирование информации

Способы представления информации в ЭВМ, кодирование и преобразование кодов в значительной степени зависят от принципа действия устройств, в которых эта информация формируется, накапливается, обрабатывается и отображается.

Целое число с основанием  $N_1$  переводится в систему счисления с основанием  $N_2$  путем последовательного деления числа  $A_{N_1}$  на основание  $N_2$ , записанного в виде числа с основанием  $N_1$ , до получения остатка. Полученное частное следует вновь делить на основание  $N_2$ , и этот процесс надо повторять до тех пор, пока частное не станет меньше делителя. Полученные остатки от деления и последнее частное записываются в порядке, обратном полученному при делении. Сформированное число и будет являться числом с основанием  $N_2$ .

**Пример 2.3.**  $A_{10}=37$   $A_2=?$   $A_{16}=?$

$$\begin{array}{r} 1) 37 \overline{) 2} \\ \underline{18} \phantom{0} \\ 09 \phantom{0} \\ \underline{14} \phantom{0} \\ 02 \phantom{0} \\ \underline{01} \phantom{0} \\ 01 \end{array}$$

$$\begin{aligned} A_{10} &= 37 \\ A_2 &= 100101 \end{aligned}$$

$$\begin{array}{r} 2) 37 \overline{) 16} \\ \underline{52} \phantom{0} \\ 52 \end{array}$$

$$\begin{aligned} A_{10} &= 37 \\ A_{16} &= 25 \end{aligned}$$

Дробное число с основанием  $N_1$  переводится в систему счисления с основанием  $N_2$  путем последовательного умножения  $A_{N_1}$  на основание  $N_2$ , записанное в виде числа с основанием  $N_1$ . При каждом умножении целая часть произведения берется в виде очередной цифры соответствующего разряда, а оставшаяся дробная часть принимается за новое множимое. Число умножений определяет разрядность полученного результата, представляющего число  $A_{N_1}$  в системе счисления  $N_2$ .

**Пример 2.4.**  $A_{10}=0,625$   $A_2=?$   $A_8=?$   $A_{16}=?$

a)

$$\begin{array}{r} 0.625 \\ \times 2 \\ \hline 1.250 \\ \times 2 \\ \hline 0.500 \\ \times 2 \\ \hline 1.000 \end{array}$$

$$A_2 = 0.101$$

b)

$$\begin{array}{r} 0.625 \\ \times 8 \\ \hline 5.000 \end{array}$$

$$A_8 = 0.5$$

c)

$$\begin{array}{r} 0.625 \\ \times 16 \\ \hline 10.000 \end{array}$$

$$A_{16} = 0.A$$

Различные виды информации могут быть разделены на две группы: статические и динамические. Так, числовая, логическая и символьная информация является статической - ее значение не связано со временем. В отличие от перечисленных типов вся аудиоинформация имеет динамический характер. Она существует только

в режиме реального времени, ее нельзя остановить для более подробного изучения. Если изменить масштаб времени (увеличить или уменьшить), аудиоинформация искажается. Это свойство иногда используется для получения звуковых эффектов.

Для кодирования символьной или текстовой информации применяются различные системы: при вводе информации с клавиатуры кодирование происходит при нажатии клавиши, на которой изображен требуемый символ, при этом в клавиатуре вырабатывается так называемый scan-код, представляющий собой двоичное число, равное порядковому номеру клавиши.

Номер нажатой клавиши никак не связан с формой символа, нанесенного на клавише. Опознавание символа и присвоение ему внутреннего кода ЭВМ производится специальной программой по специальным таблицам: ДКОИ, КОИ-7, ASCII (Американский стандартный код передачи информации).

Всего с помощью таблицы кодирования ASCII можно закодировать 256 различных символов. Эта таблица разделена на две части: основную (с кодами от 00h до 7Fh) и дополнительную (от 80h до FFh, где буква h обозначает принадлежность кода к шестнадцатеричной системе счисления).

Таблица кодирования текстовой информации ASCII

Radix : Hex															
0	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
р	q	г	с	д	е	ф	г	х	у	z	{		}	~	
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
А	Б	В	Г	Д	Е	Ж	З								

$$A_n = a_{m-1}a_{m-2}\dots a_1a_0 \cdot a_{-1}a_{-2}\dots a_{-k} = a_{m-1} \cdot N^{m-1} + a_{m-2} \cdot N^{m-2} + \dots + a_{-k} \cdot N^{-k}$$

$$A_n = \sum_{i=-k}^{m-1} a_i \cdot N^i$$

где  $a_i$  –  $i$ -я цифра числа;  $k$  – количество цифр в дробной части числа;  $m$  – количество цифр в целой части числа;  $N$  – основание системы счисления.

Основание системы счисления  $N$  показывает, во сколько раз “вес”  $i$ -го разряда больше  $(i-1)$  разряда. Целая часть числа отделяется от дробной части точкой (запятой).

$$A_{10} = 37.25.$$

В соответствии с формулой это число формируется из цифр с весами рядов:

$$A_{10} = 3 \cdot 10^1 + 7 \cdot 10^0 + 2 \cdot 10^{-1} + 5 \cdot 10^{-2}.$$

Теоретически наиболее экономичной системой счисления является система с основанием  $e = 2,71828\dots$ , находящимся между числами 2 и 3.

Во всех современных ЭВМ для представления числовой информации используется двоичная система счисления. Это обусловлено:

более простой реализацией алгоритмов выполнения арифметических и логических операций;

более надежной физической реализацией основных функций, так как они имеют всего два состояния (0 и 1);

экономичностью аппаратной реализации всех схем ЭВМ.

При  $N=2$  число различных цифр, используемых для записи чисел, ограничено множеством из двух цифр (ноль и единица). Кроме двоичной системы счисления широкое распространение получили и производные системы:

двоичная -  $\{0,1\}$ ;

десятичная, точнее двоично-десятичное представление десятичных чисел, -  $\{0,1,\dots,9\}$ ;

шестнадцатеричная -  $\{0,1,2, \dots, 9, A, B, C, D, E, F\}$ . Здесь шестнадцатеричная цифра  $A$  обозначает число 10,  $B$ -число 11, ...,  $F$ -число 15;

восьмеричная (от слова восьмерик) -  $\{0,1,2,3,4,5, 6, 7\}$ . Она широко используется во многих специализированных ЭВМ.

Восьмеричная и шестнадцатеричная системы счисления являются производными от двоичной, так как  $16 = 2^4$  и  $8 = 2^3$ . Они используются в основном для более компактного изображения двоичной информации, так как запись значения чисел производится существенно меньшим числом знаков.

Число в двоичной, восьмеричной и шестнадцатеричной системах счисления имеет следующее представление:

$$A_2 = 1100100,101;$$

$$A_8 = 144.5;$$

$$A_{16} = 64.A;$$

$$A_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3};$$

$$A_8 = 1 \cdot 8^2 + 4 \cdot 8^1 + 4 \cdot 8^0 + 5 \cdot 8^{-1};$$

$$A_{16} = 6 \cdot 16^1 + 4 \cdot 16^0 + 10 \cdot 16^{-1}.$$

Представление чисел в различных системах счисления допускает однозначное преобразование их из одной системы в другую. В ЭВМ перевод из одной системы в другую осуществляется автоматически по специальным программам. Правила перевода целых и дробных чисел отличаются.

## **Раздел 1. Общая характеристика процессов сбора, передачи, обработки и накопления информации**

### **Тема 1.4. Понятие файла**

Информация, представляемая для обработки на компьютере, называется данными. Для хранения на устройствах внешней памяти данные организуют в виде файлов. Файл — именованная область внешней памяти.

Способ организации как служебной, так и пользовательской информации о файлах на носителях называют файловой системой. Конкретная файловая система определяет, в частности, правила именования файлов.

Необходимые для выполнения операций с файлами и носителями программные средства входят в состав операционных систем. Такие программные средства не изменяют и не обращаются к содержимому файлов, а оперируют с ними просто как с целым, непрерывным массивом данных. Таким образом, файловая система обеспечивает выполнение операций для любых программ.

Имя файлу присваивает пользователь, или программа, создающая файл, предлагает имя в автоматическом режиме. По историческим причинам для пользователя имя файла в операционных системах фирмы Майкрософт состоит из двух частей, разделенных точкой: собственно имени и расширения. Тип файла определяется по его расширению, которое задает программа, сохраняющая файл.

С точки зрения прикладных программ, файл представляет собой некоторую последовательность байтов. Используя такой подход, как доступ к файлам, организуется также доступ к некоторым устройствам, которые принимают или возвращают поток байтов. К таким устройствам относятся принтеры, модемы, клавиатура или поток текстового вывода на экран и др.

В некоторых операционных системах предусмотрен такой доступ и к служебной информации самих носителей. Для работы с такими файлами предусмотрены специальные, зарезервированные системой, имена файлов.

Следует помнить, что для ОС линии Microsoft:

между именем и расширением ставится точка, не входящая ни в имя, ни в расширение;  
имя файла можно набирать в любом регистре, т.к. для системы все буквы строчные;  
символы, не используемые в имени файла \* = + [ ] \ ; : , . < > / ? ;  
имена устройств не могут использоваться в качестве имён файлов (prn, lpt, com, con, nul ).

Наиболее часто встречающиеся расширения:

EXE, COM — готовая к выполнению программа;  
BAT — пакетный командный файл;  
SYS — программа-драйвер устройства (системная);  
BAK — резервная копия файла;  
OBJ — объектный модуль («полуфабрикат» программы);  
DAT — файл данных со служебной информацией;  
BAS — исходный текст программы на Бейсике;  
TXT — текстовый файл;  
DOC — документ, созданный в Microsoft Word.

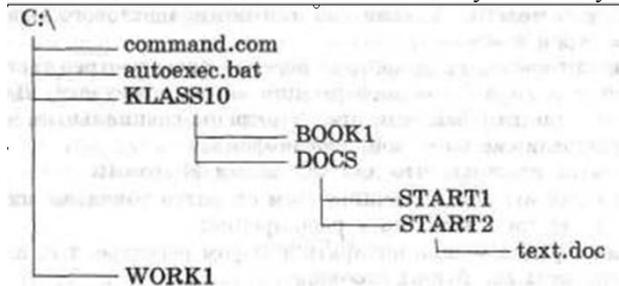
### Каталоги и маршруты

Для удобства хранения и работы файловые структуры организуются с помощью вложенных каталогов (папок).

Каталог — специальный системный файл, в котором хранится служебная информация о файлах. На каждом носителе может быть множество каталогов. В каждом каталоге может быть зарегистрировано много файлов, но каждый файл регистрируется только в одном каталоге.

На каждом логическом томе присутствует один главный, или корневой, каталог. В нем регистрируются файлы и подкаталоги (каталоги 1 уровня). В каталогах 1 уровня регистрируются файлы и каталоги 2 уровня и т. д. Получается древовидная структура каталогов, например:

Каталог, с которым работает пользователь в настоящий момент, называется текущим. Когда используется файл не из текущего каталога, программе, осуществляющей доступ к файлу, необходимо указать, где именно этот файл находится. Это делается с помощью указания пути к файлу.



Путь к файлу — это последовательность имен каталогов, в операционных системах Windows разделенных символом «\» (в ОС линии UNIX используется символ «/»). Этот путь задает маршрут к тому каталогу, в котором находится нужный файл.

Рассмотрим, например, запись  
\\KCLASS10\DOCS\START2\text.doc

Она означает, что файл text.doc находится в подкаталоге START2, который находится в каталоге DOCS, а он в свою очередь находится в каталоге KCLASS10 корневого каталога. Если перед первым символом «обратный слэш» появится точка, то отсчет будет вестись от текущего каталога. Каталог «две точки» — это каталог, в который входит текущий.

### Работа с файлами

Над файлами можно производить следующие основные операции: копирование, перемещение, удаление, переименование и пр. Каждый файл на диске имеет свой адрес. Чтобы понять принцип доступа к информации, хранящейся в файле, необходимо знать способ записи данных на носители информации.

Перед использованием диск размечается на дорожки и секторы (форматируется). С точки зрения оборудования разметка — это процесс записи на носитель служебной информации, отмечающей конец и начало каждого сектора. Обычный объем сектора — 512 байт. На одной стороне размещается 80 дорожек. Каждая дорожка содержит 18 секторов.

Названия «сектор», «дорожка» введены для дисковых носителей. Во многих современных носителях информации, использующих хранение данных в энергонезависимой памяти, эти понятия поддерживаются реализацией файловых систем для обеспечения общих принципов работы.

В одной из распространенных файловых систем, FAT, предусматривается, что все файлы перечислены в каталогах. Обязателен корневой каталог, размещенный в определенном месте диска. О каждом из перечисленных в каталоге файлов помимо обычных данных известно местоположение (в виде номера) начала файла.

Для того, чтобы определить, какие именно секторы занимает файл, применяется второй обязательный элемент файловой системы — таблица FAT (размещения файлов). Таблица представляет собой массив ячеек. Размер ячейки фиксирован и отражается в номере файловой системы (12, 16, 32 бита). Каждый файл занимает

некоторую последовательность секторов, не обязательно последовательно расположенных. При сохранении файла в ячейку записывается номер следующего сектора в цепочке.

Поскольку на современных дисках секторов существенно больше, чем можно записать номеров в таблице, то секторы объединяют в кластеры. Именно кластерами и распределяется пространство на дисках, в результате эта файловая система неэффективно работает с мелкими файлами.

Сделать эту проблему менее острой позволяет увеличение размера ячейки в FAT. Это позволяет уменьшить размер кластера и увеличить количество адресов (файлов) на диске. В операционных системах, начиная с Windows 98, реализована FAT-32.

Помимо этой файловой системы, существует большое количество других, разработанных для разных операционных систем и решаемых задач.

## Раздел 2. Локальные и глобальные сети

### Тема 2.1. Сети ЭВМ

По мере удешевления процессоров в начале 60-х годов появились новые способы организации вычислительного процесса. Начали развиваться интерактивные многотерминальные системы разделения времени, ставшие прообразом современных компьютерных сетей. Они уже имели все внешние признаки локальных вычислительных сетей, однако по существу ими не являлись, так как обработка данных автономно работающего компьютера выполнялась централизованно. Кроме того, потребность предприятий в создании локальных сетей в это время еще не созрела - в одном здании просто нечего было объединять в сеть, так как из-за высокой стоимости вычислительной техники предприятия не могли себе позволить роскошь приобретения нескольких компьютеров.

Исторически первыми появились глобальные сети. Они создавались с целью обеспечения возможности обмена данными на больших расстояниях между менфреймами и супер ЭВМ.

По мере удешевления компьютеров, их количество начало стремительно возрастать, и следующим шагом на пути повышения эффективности работы фирмы (предприятия) явилось создание в 60-х годах локальных вычислительных сетей. Каждая из них представляла собой систему вычислительных машин одной организации или фирмы, расположенной в одном или нескольких близлежащих зданиях. Основной целью создания локальных сетей первоначально было разделение локальных ресурсов сети (дорогостоящих) между всеми пользователями. В настоящее время наряду с предоставлением сетевых ресурсов актуальным является возможность обмена различной информацией между пользователями сети (документооборот, on-line игры, общение и др.).

Появление всё большего количества ЛВС способствовало развитию и глобальных сетей, логическим завершением чего стало появление глобальной сети Internet. Она связала в единую систему большое количество частных, коммерческих, государственных, академических и других информационных сетей по всему миру, то есть стала сетью компьютерных сетей.

При физическом соединении двух и более компьютеров образуется компьютерная сеть. В общем случае необходимо аппаратное обеспечение (сетевое) и программы работы с сетью.

Одной из первых задач, возникших при развитии вычислительной техники, стала задача обеспечения значительно большей надежности управления процессом, чем могла обеспечить одна ЭВМ, например, управление космическим аппаратом. Другим фактором, оказавшим существенное влияние на развитие локальных сетей, стало создание автоматизированных систем управления (АСУ).

Любая сеть подразумевает её администрирование, т.е. управление сетевыми политиками, т.е. совокупностью приёмов разделения и ограничения прав участников сети.

Компьютерная сеть представляет собой достаточно сложную систему (рис.) и характеризуется топологией, протоколами, интерфейсами, сетевыми техническими и программными средствами.

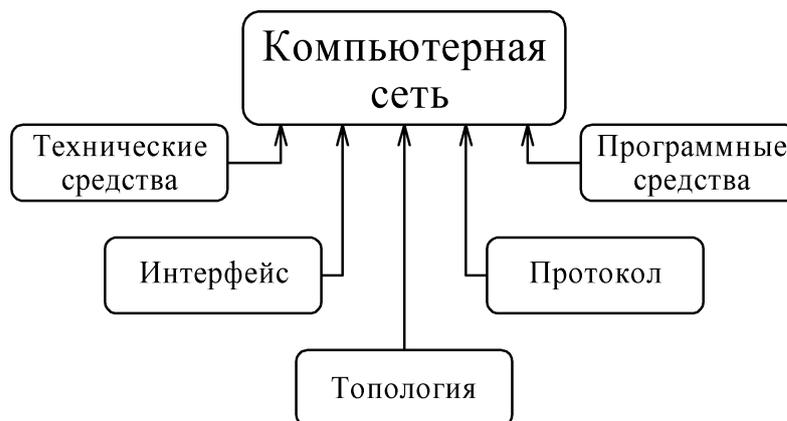


Рис. Состав компьютерной сети

Топология компьютерной сети отражает структуру связей между её основными функциональными элементами. В зависимости от рассматриваемых компонентов, принято различать физическую и логическую структуры локальных сетей. Физическая структура определяет топологию физических соединений между компьютерами. Логическая структура определяет логическую организацию взаимодействия компьютеров между

собой. Дополняя друг друга, физическая и логическая структуры дают более полное представление о компьютерной сети.

Под сетевыми техническими средствами подразумеваются различные физические устройства, обеспечивающие объединение компьютеров в единую компьютерную сеть.

Протоколы представляют собой правила взаимодействия функциональных элементов сети. Интерфейсы – средства сопряжения функциональных элементов сети. Следует обратить внимание, что в качестве функциональных элементов могут выступать как отдельные устройства, так и программные интерфейсы.

Сетевые программные средства осуществляют управление работой компьютерной сети и обеспечивают соответствующий интерфейс с пользователями. К сетевым программным средствам относятся сетевые операционные системы и вспомогательные (сервисные) программы.

Значимые практические результаты по объединению компьютеров в сети были получены в конце 60-х, когда с помощью глобальных связей и техники коммутации пакетов удалось реализовать взаимодействие машин класса мэйнфреймов и суперкомпьютеров. Эти дорогостоящие компьютеры хранили уникальные данные и программы, обмен которыми позволил повысить эффективность их использования. Эти сети стали первыми и получили название территориально-распределенные (глобальные) сети (WAN – Wide Area Networks).

Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, в первых глобальных сетях часто использовались уже существующие телефонные каналы связи – для экономии затрат на создание сети. Поскольку скорость передачи данных по таким каналам была низкой, набор предоставляемых услуг в глобальных сетях такого типа ограничивался передачей файлов и электронной почтой. Для глобальной сети важно было само наличие связи между далеко удаленными друг от друга компьютерами. Другой недостаток заключался в значительных искажениях передаваемых сигналов. Поэтому протоколы глобальных сетей, построенных с использованием каналов связи низкого качества, отличаются сложными процедурами контроля и восстановления данных. Развитие технологии глобальных компьютерных сетей во многом определялось прогрессом телефонных сетей. С конца 60-х годов в телефонных сетях все чаще стала применяться передача голоса в цифровой форме, что привело к появлению высокоскоростных цифровых каналов.

В начале 70-х годов произошел технологический прорыв в области производства компьютерных компонентов – появились большие интегральные схемы. Их сравнительно невысокая стоимость и высокие функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов. Потребности пользователей вычислительной техники росли, им стало недостаточно собственных компьютеров, им уже хотелось получить возможность обмена данными с другими близко расположенными компьютерами. В ответ на эту потребность предприятия и организации стали соединять свои мини-компьютеры вместе и разрабатывать программное обеспечение, необходимое для их взаимодействия. В результате появились первые локальные сети (LAN – Local Area Networks).

В середине 80-х годов положение дел в локальных сетях стало кардинально меняться. Утвердились стандартные технологии объединения компьютеров в сеть - Ethernet, Arcnet, Token Ring и др. Мощным стимулом для развития послужили персональные компьютеры. Эти массовые продукты явились идеальными элементами для построения сетей - с одной стороны, они были достаточно мощными для работы сетевого программного обеспечения, а с другой – явно нуждались в объединении своей вычислительной мощности для решения сложных задач, а также разделения дорогих периферийных устройств и дисковых массивов.

Современные LAN – это сети, сосредоточены на небольшой территории (1-2 км) и построены с использованием дорогих высококачественных линий связи, что позволяет достигать больших скоростей передачи данных (до 1 Гбит/с) и использовать более простые сетевые протоколы.

Последними появились городские сети (MAN – Metropolitan Area Networks). Они предназначены для обслуживания территории крупного города (мегаполиса). В то время как локальные сети наилучшим образом подходят для разделения ресурсов на коротких расстояниях, а широкощелевые передачу, а глобальные сети обеспечивают работу на больших расстояниях, но с ограниченной скоростью и небогатым набором услуг, сети мегаполисов занимают некоторое промежуточное положение. Они используют цифровые магистральные линии связи, часто оптоволоконные, и предназначены для связи локальных сетей в масштабах города и соединения локальных сетей с глобальными.

Сегодня вычислительные сети продолжают развиваться, причем достаточно быстро. Разрыв между локальными и глобальными сетями постоянно сокращается во многом из-за появления высокоскоростных территориальных каналов связи, не уступающих по качеству кабельным системам локальных сетей. В них стала обрабатываться несвойственная ранее вычислительным сетям информация – голос, видеоизображения, рисунки.

## **Раздел 2. Локальные и глобальные сети**

### **Тема 2.2. Мировая компьютерная сеть**

После Карибского кризиса в США сформулирована задача создания коммуникационной сети, устойчивой к частичным отказам. Цель – координация действий, поддержание связи между военными базами США. При внезапном уничтожении любого сегмента сети она должна продолжать функционирование. В середине 70-х годов создается первая глобальная вычислительная сеть ARPAnet (Advanced Research Projects Agency – управление перспективных исследований министерства обороны США), отвечающая поставленным условиям. Она объединяет военные базы и исследовательские центры МО в США и за рубежом. С 1986 году была создана высокоскоростная сеть NSFNET, объединившая 5 суперкомпьютеров США, расположенными в различных научных центрах. Она стала основой для развития сети Internet.

Internet — это общемировая совокупность сетей, связывающая между собой миллионы компьютеров, причем число пользователей Internet лавинообразно увеличивается. С технической точки зрения это объедине-

ние транснациональных компьютерных сетей, связывающих всевозможные типы компьютеров, физически передающих данные всеми доступными типами носителей: по телефонным проводам, оптоволокну, через спутниковые каналы.

Для каждого компьютера и Internet устанавливаются два адреса: цифровой IP-адрес (IP – Internet Protocol – межсетевой протокол) и доменный адрес. Оба эти адреса могут применяться равноценно. Цифровой адрес удобен для обработки на компьютере, а доменный для восприятия пользователем. Цифровой адрес состоит из четырех десятичных чисел – каждое в диапазоне от 0 до 255 (1 байт двоичного кода) и содержит полную информацию, необходимую для идентификации компьютера. IP-адрес включает в себя адрес сети, адрес подсети, адрес компьютера в сети. При работе в сети машины отыскивают друг друга именно по этим числам. Числа записываются через точку, например: 175.94.93.09. Доменный адрес состоит из осмысленных буквенных обозначений. Синтаксис записи доменного адреса таков: протокол://имя машины.имя домена [каталог/ подкаталог/имя файла].

Если рассматривать сеть Internet на нижнем уровне, то она позволяет пересылать пакеты данных с одного подключенного к Internet компьютеру на любой другой. Маршрутизаторы Internet автоматически выбирают оптимальный маршрут для доставки пакетов данных до места назначения. На этой основе были разработаны различные службы, использующие возможности Internet. До середины 90-х годов в основном применялись: электронная почта, телеконференции, серверы новостей, файловые серверы.

Пока польза от Internet ограничивалась возможностью применения перечисленных выше служб, Internet использовался в основном в научной среде, главным образом, для электронной почты. В 1993-1994 г. ситуация в корне переменялась. Причиной этого стало появление и широкое внедрение в сети Internet новой службы (подсистемы) — World Wide Web, в буквальном переводе — всемирной паутины (сокращенно: WWW или Web). С этого момента количество пользователей Internet стало лавинообразно увеличиваться — в несколько раз за год.

WWW — это содержащаяся в Internet всемирная распределенная база гипертекстовых документов. Каждый компьютер, имеющий постоянное подключение в Internet, можно использовать в качестве Web-сервера (или даже нескольких Web-серверов). Любой пользователь, имеющий диалоговое соединение к Internet, может общаться с другим пользователем (группой пользователей) в реальном времени, проводить видеотелеконференции и т.д. Пользователь может соединиться с любым Web-сервером с помощью Web-браузера. Для этого необходимо указать только электронный адрес Web-сервера. WWW работает по принципу «клиент-серверы»: существует множество серверов, которые по запросу клиента возвращают ему интересующую информацию. Каждый желающий может создать собственные WWW-странички, и разместить на существующем Web-сервере. Фирмы публикуют в Web сведения о своей деятельности, продукции, техническую документацию и т.д. На WWW-серверах университетов можно прочесть научные труды их сотрудников, сведения о приеме и выпуске студентов, учебные планы и т.д.

Формально не существует органа, управляющего Internet (как нет и единой сети Internet, а есть совокупность отдельных сетей). Направления развития Internet определяет ISOC (Internet Society) – общественная организация, работающая на принципах добровольного и безвозмездного участия энтузиастов. ISOC выбирает "совет старейшин" IAB (Internet Architecture Board – совет по архитектуре Internet), который утверждает стандарты, распределяет ресурсы (адреса) и решает проблемы по мере их появления. Решением технических и организационных проблем, выпуском документации и формированием предложений для IAB по изменению старых стандартов или введению новых занимается IETF (Internet Engineering Task Force – инженерная комиссия).

Протокол – в широком смысле – специальный стандарт (набор норм, договорённостей и правил), предназначенный для обеспечения совместимости работы компьютеров на аппаратном и программном уровнях. В сетях и коммуникационных системах под протоколом понимают формальную спецификацию, описывающую процедуры, выполняемые при получении данных. Протокол описывает формат, методы синхронизации и последовательность передачи данных по сети, а также методы проверки на наличие ошибок.

По мере продвижения пакета данных по сети на каждом этапе его взаимодействия с другими сетевыми элементами обрабатываются протоколы разных уровней. Полную совокупность таких протоколов, необходимых для успешного взаимодействия разных элементов в рамках сети данного типа, принято называть семейством. Интернет работает под семейством протоколов TCP/IP, который обеспечивает функционирование и доступ к сети. Семейство протоколов TCP/IP, включающий в себя большое количество протоколов различного уровня и назначения и отвечающий за передачу информации, проходящей по сети и её приём. Непосредственно сам протокол TCP/IP состоит из двух протоколов: TCP – протокол транспортного уровня – управляет тем, как происходит передача информации; IP – адресный протокол – принадлежит сетевому уровню и определяет, куда происходит передача.

TCP (Transmission Control Protocol) протокол нарезает данные на пакеты, после чего каждый пакет маркируется таким образом, чтобы в нем были данные, необходимые для правильной сборки документа на компьютере получателя.

IP (Internet Protocol) протокол обеспечивает управление перемещением пакетов (путём трассировки межсетевых адресов), а также маршрутизацию исходящих сообщений и распознавание входящих. У каждого участника всемирной сети должен быть свой уникальный адрес (IP адрес). Этот адрес выражается четырьмя байтами (Например, IP-адрес: 192.168.001.110). Каждая ЭВМ, подключенная к сети интернет, имеет уникальный физический адрес, состоящий из десятичных чисел, отделенных точкой. Каждое число изменяется от 0 до 255, что даёт число пользователей интернет  $256^4 = 4,3 \text{ ёддä}$ . Общее количество адресов может реально составлять

около 2 млрд. Теоретическое количество адресов в 2 раза выше, однако существуют служебные и резервные адреса, а также ограничения, накладываемые особенностями организации сети.

## **Раздел 2. Локальные и глобальные сети**

### **Тема 2.3. Вредоносное программное обеспечение**

К вредоносному программному обеспечению относятся сетевые черви, классические файловые вирусы, троянские программы, хакерские утилиты и прочие программы, наносящие заведомый вред компьютеру, на котором они запускаются на выполнение, или другим компьютерам в сети.

#### **Сетевые черви**

К данной категории относятся программы, распространяющие свои копии по локальным и/или глобальным сетям с целью:

- проникновения на удаленные компьютеры;
- запуска своей копии на удаленном компьютере;
- дальнейшего распространения на другие компьютеры в сети.

Для своего распространения сетевые черви используют разнообразные компьютерные и мобильные сети: электронную почту, системы обмена мгновенными сообщениями, файлообменные (P2P) и IRC-сети, LAN, сети обмена данными между мобильными устройствами (телефонами, карманными компьютерами) и т. д.

Большинство известных червей распространяется в виде файлов: вложение в электронное письмо, ссылка на зараженный файл на каком-либо веб- или FTP-ресурсе в ICQ- и IRC-сообщениях, файл в каталоге обмена P2P и т. д.

Некоторые черви (так называемые «бесфайловые» или «пакетные» черви) распространяются в виде сетевых пакетов, проникают непосредственно в память компьютера и активизируют свой код.

Для проникновения на удаленные компьютеры и запуска своей копии черви используют различные методы: социальный инжиниринг (например, текст электронного письма, призывающий открыть вложенный файл), недочеты в конфигурации сети (например, копирование на диск, открытый на полный доступ), ошибки в службах безопасности операционных систем и приложений.

Некоторые черви обладают также свойствами других разновидностей вредоносного программного обеспечения. Например, некоторые черви содержат троянские функции или способны заражать выполняемые файлы на локальном диске, т. е. имеют свойство троянской программы и/или компьютерного вируса.

#### **Классические компьютерные вирусы**

К данной категории относятся программы, распространяющие свои копии по ресурсам локального компьютера с целью:

- последующего запуска своего кода при каких-либо действиях пользователя;
- дальнейшего внедрения в другие ресурсы компьютера.

В отличие от червей, вирусы не используют сетевых сервисов для проникновения на другие компьютеры. Копия вируса попадает на удаленные компьютеры только в том случае, если зараженный объект по каким-либо не зависящим от функционала вируса причинам оказывается активизированным на другом компьютере, например:

- при заражении доступных дисков вирус проник в файлы, расположенные на сетевом ресурсе;
- вирус скопировал себя на съёмный носитель или заразил файлы на нем;
- пользователь отослал электронное письмо с зараженным вложением.

Некоторые вирусы содержат в себе свойства других разновидностей вредоносного программного обеспечения, например бэкдор-процедуру или троянскую компоненту уничтожения информации на диске.

#### **Троянские программы**

В данную категорию входят программы, осуществляющие различные несанкционированные пользователем действия: сбор информации и ее передачу злоумышленнику, ее разрушение или злонамеренную модификацию, нарушение работоспособности компьютера, использование ресурсов компьютера в неблагоприятных целях.

Отдельные категории троянских программ наносят ущерб удаленным компьютерам и сетям, не нарушая работоспособность зараженного компьютера (например, троянские программы, разработанные для массированных DoS-атак на удаленные ресурсы сети).

#### **Хакерские утилиты и прочие вредоносные программы**

К данной категории относятся:

- утилиты автоматизации создания вирусов, червей и троянских программ (конструкторы);
- программные библиотеки, разработанные для создания вредоносного ПО;
- хакерские утилиты скрытия кода зараженных файлов от антивирусной проверки (шифровальщики файлов);

«злые шутки», затрудняющие работу с компьютером;

программы, сообщающие пользователю заведомо ложную информацию о своих действиях в системе;

прочие программы, тем или иным способом намеренно наносящие прямой или косвенный ущерб данному или удаленным компьютерам.

#### **Методы защиты данных**

Основным видом угроз целостности и конфиденциальности информации являются преднамеренные угрозы, заранее планируемые злоумышленниками для нанесения вреда. Их можно разделить на две группы:

- угрозы, реализация которых выполняется при постоянном участии человека;
- угрозы, реализация которых после разработки злоумышленником соответствующих компьютерных программ выполняется этими программами без непосредственного участия человека.

Задачи по защите от угроз каждого вида одинаковы:  
запрещение несанкционированного доступа к ресурсам вычислительных систем;  
невозможность несанкционированного использования компьютерных ресурсов при осуществлении доступа;  
своевременное обнаружение факта несанкционированных действий, устранение их причин и последствий.

Основным способом запрещения несанкционированного доступа к ресурсам вычислительных систем является подтверждение подлинности пользователей и разграничение их доступа к информационным ресурсам, включающего следующие этапы:

идентификация необходима для указания компьютерной системе уникального идентификатора обращающегося к ней пользователя. Идентификатор может представлять собой любую последовательность символов и должен быть заранее зарегистрирован в системе администратора службы безопасности. В процессе регистрации заносится следующая информация:

фамилия, имя, отчество (при необходимости другие характеристики пользователя);  
уникальный идентификатор пользователя;  
имя процедуры установления подлинности;  
эталонная информация для подтверждения подлинности (например, пароль);  
ограничения на используемую эталонную информацию (например, время действия пароля);  
полномочия пользователя по доступу к компьютерным ресурсам.  
установление подлинности (аутентификация) заключается в проверке истинности полномочий пользователя.  
определение полномочий для последующего контроля и разграничения доступа к компьютерным ресурсам.

Для особо надежного опознания при идентификации используются технические средства, определяющие индивидуальные характеристики человека (голос, отпечатки пальцев, структура зрачка).

Однако такие методы требуют значительных затрат и поэтому используются редко. Наиболее массово используемыми являются парольные методы проверки подлинности пользователей. Пароли можно разделить на две группы:

1. Простой пароль не изменяется от сеанса к сеансу в течение установленного периода его существования.

Динамически изменяющийся пароль изменяется по правилам, определяемым используемым методом. Выделяют следующие методы реализации динамически изменяющихся паролей:

методы модификации простых паролей. Например, случайная выборка символов пароля и одноразовое использование паролей;

метод «запрос—ответ», основанный на предъявлении пользователю случайно выбираемых запросов из имеющегося массива;

функциональные методы, основанные на использовании не которой функции  $F$  с динамически изменяющимися параметрами (дата, время, день недели и др.), с помощью которой определяется пароль.

Для защиты от несанкционированного входа в компьютерную систему используются как общесистемные, так и специализированные программные средства защиты.

После идентификации и аутентификации пользователя система защиты должна определить его полномочия для последующего контроля санкционированного доступа к компьютерным ресурсам (разграничение доступа). В качестве компьютерных ресурсов рассматриваются:

программы;  
внешняя память (файлы, каталоги, логические диски);  
информация, разграниченная по категориям в базах данных;  
оперативная память;  
время (приоритет) использования процессора;  
порты ввода-вывода;  
внешние устройства.

Различают следующие виды прав пользователей по доступу к ресурсам:

всеобщее (полное предоставление ресурса);  
функциональное или частичное;  
временное.

Наиболее распространенными способами разграничения доступа являются:  
разграничение по спискам (пользователей или ресурсов);  
использование матрицы установления полномочий (строки матрицы — идентификаторы пользователей, столбцы — ресурсы компьютерной системы);  
разграничение по уровням секретности и категориям (например, общий доступ, конфиденциально, секретно);  
парольное разграничение.

### **Раздел 3. Технические средства организации информационных процессов**

#### **Тема 3.1. Основы устройства ЭВМ**

Еще при создании первых компьютеров в 1945 году знаменитый математик Джон фон Нейман описал, как должен быть устроен компьютер, чтобы он был универсальным и эффективным устройством для обработки

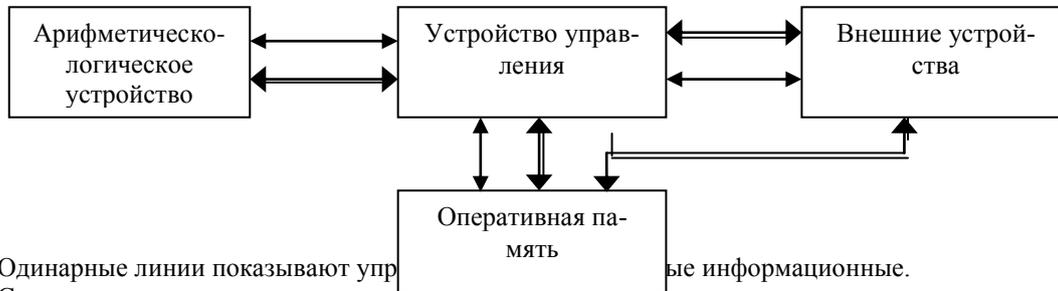
информации. Эти основы конструкции компьютера называются принципами Фон Неймана. Сейчас подавляющее большинство компьютеров в основных чертах соответствует принципам Фон Неймана.

Устройство компьютера. Согласно принципам Фон Неймана, компьютер должен иметь следующие устройства:

- Арифметическо-логическое устройство, выполняющие арифметические и логические операции;
- Устройство управления, которое организует процесс выполнения программ;
- Внешние устройства для ввода-вывода информации.

Память компьютера должна состоять из некоторого количества пронумерованных ячеек, в каждой из которых могут находиться или обрабатываться данные, или инструкции программ. Все ячейки памяти должны быть одинаково легко доступны для других устройств компьютера.

Связи между устройствами компьютера.



Одинарные линии показывают управление, двойные — информационные.

Следует заметить, что схема устройств современных компьютеров несколько отличается от приведенных выше. В частности, арифметическо-логическое устройство и устройство управления, как правило объединены в единое устройство — центральный процессор. Кроме того, процесс выполнения программ может прерываться для выполнения неотложных действий, связанных с поступившими сигналами от внешних устройств компьютера — прерываний.

### Поколения ЭВМ

В истории вычислительной техники существует своеобразная периодизация ЭВМ по поколениям. В ее основу первоначально был положен физико-технологический принцип: машину относят к тому или иному поколению в зависимости от используемых в ней физических элементов или технологии изготовления. Границы поколений во времени размыты, так как в одно и то же время выпускались машины совершенно разного уровня.

В настоящее время физико-технологический принцип не является единственным при определении принадлежности той или иной ЭВМ к поколению. Следует рассматривать также и уровень программного обеспечения, быстродействие, другие факторы, основные из которых сведены в табл.

Таблица – Поколения ЭВМ

Показатель	Поколения ЭВМ					
	Первое	Второе	Третье	Четвертое		Пятое
				А	В	
Выпуск первых ЭВМ	1951-1954	1958-1960	1965-1966	1976-1979	С 1985	
Элементная база процессора	Электронные лампы	Транзисторы	Интегральные схемы (ИС)	Большие ИС (БИС)	Сверхбольшие ИС (СБИС)	+Оптоэлектроника + криоэлектроника
Элементная база ОЗУ	Электронно-лучевые трубки	Ферритовые сердечники	Ферритовые сердечники	БИС	СБИС	СБИС
Максимальная емкость ОЗУ, байт	102	103	104	105	107	108
Максимальное быстродействие процессора, опер/с	104	106	107	108	109 +	1012+
Языки программирования	Машинный код	+ Ассемблер	+ Процедурные языки высокого уровня (ЯВУ)	+ Новые процедурные ЯВУ	+ Не процедурные ЯВУ	+ Новые Не процедурные ЯВУ

Сред- ства связи пользователя с ЭВМ	Пуль- т управ- ления и перфо- карты	Пер- фокар-ты и пер- фоленты	Алфа- вит-но-циф- ровой тер- минал	Мо- но- хромный графичес- кий дис- плей, кла- виа- тура	Цвет- ной графиче- ский дис- плей, клави- атура, мышь и др.	+ Устрой- ства голосовой связи с ЭВМ
--	---	---------------------------------------	---	--	--	---

Следует понимать, что разделение ЭВМ по поколениям, весьма относительно. Первые ЭВМ были «штучными» изделиями, на которых отрабатывались основные принципы, нет оснований относить их к какому-либо поколению. Нет единого мнения и при определении признаков пятого поколения. В середине 1980-х гг. считалось, что основной признак этого (будущего) поколения – полновесная реализация принципов искусственного интеллекта. Эта задача оказалась значительно сложнее, чем виделось в то время, и ряд специалистов снижают требования к этому этапу (и даже утверждают, что он уже состоялся).

В 1990-х гг. микроэлектроника подошла к пределу, разрешенному физическими законами. Высока плотность упаковки компонентов в интегральных схемах и почти предельно велика предельная скорость их работы.

В совершенствовании будущих ЭВМ видны два пути. На физическом уровне это переход к использованию иных физических принципов построения узлов ЭВМ – на основе оптоэлектроники, использующей оптические свойства материалов, на базе которых создаются процессор и оперативная память, и криогенной электроники, использующей сверхпроводящие материалы при очень низких температурах. На уровне совершенствования интеллектуальных способностей машин, не всегда определяемых физическими принципами их конструкций, постоянно возникают новые результаты, опирающиеся на принципиально новые подходы к программированию. Создание новейших информационных технологий, систем искусственного интеллекта, баз знаний, экспертных систем продолжается.

### **Раздел 3. Технические средства организации информационных процессов**

#### **Тема 3.2. ПЭВМ, ее устройство и основные технические характеристик**

Обычно ПЭВМ или ПК состоят из трех частей:

Системного блока.

Клавиатуры, позволяющей вводить символы в компьютер.

Монитора (дисплея) – для изображения текстов и графической информации.

Компьютеры выпускаются и в портативном варианте – обычно в блокнотном исполнении (ноутбук). Здесь системный блок, монитор и клавиатура, заключены в один корпус: системный блок спрятан под клавиатурой, а монитор сделан как крышка к клавиатуре.

Системный блок. Хотя из этих частей компьютера системный блок выглядит наименее эффектно, именно он является в компьютере “главным”. В нем располагаются все основные узлы компьютера:

Электронные схемы, управляющие работой компьютера (микропроцессор, оперативная память, контроллеры устройств и т.д.)

Блок питания, который преобразует электропитание сети в постоянный ток низкого напряжения, подаваемый на электронные схемы компьютера.

Накопители (дисководы) для гибких магнитных дисков, используемые для чтения и записи на гибкие магнитные диски.

Накопитель на жестком магнитном диске, предназначенный для чтения и записи на несъемный жесткий магнитный диск (винчестер).

Другие устройства.

Дополнительные устройства.

К системному блоку компьютера можно подключить различные устройства ввода-вывода информации, расширяя тем самым его функциональные возможности.

Внешние устройства. Многие устройства располагаются вне системного блока компьютера и присоединяются к нему через специальные гнезда (разъемы), находящиеся обычно на задней стенке системного блока. Такие устройства обычно называются внешними. Кроме монитора и клавиатуры такими устройствами являются.

Принтер – для вывода на печать графической и текстовой информации.

Мышь – устройство облегчающее ввод информации в компьютер.

Джойстик – манипулятор в виде укрепленной на шарнире ручки с кнопкой, употребляется в основном для компьютерных игр;

А также другие устройства.

Внутренние устройства. Некоторые устройства могут вставляться внутрь системного блока, поэтому они называются внутренними, например:

Модем или факс-модем – для обмена информацией с другими компьютерами через телефонную сеть (факс-модем может также получать и принимать факсы).

Дисковод для компакт-дисков, он обеспечивает возможность чтения данных с компьютерных компакт-дисков и проигрывание аудио компакт-дисков

Стример – для хранения данных на магнитной ленте

Звуковая карта – для воспроизведения и записи звуков(музыки, голоса и т.д.)

Впрочем, модемы, факс-модемы, стримеры и другие устройства могут выпускаться и во внешнем исполнении. Как правило, устройства во внутреннем исполнении стоят дешевле – для них не надо изготавливать корпус и их не надо снабжать своим блоком питания.

Контроллеры и устройства. Для управления работой устройств в компьютере используются электронные схемы – контроллеры. Различные устройства используют разные способы подключения к контроллерам:

Дисковод для дискет, клавиатура и т.д. подключается к имеющимся в составе компьютера стандартным контроллерам;

Звуковые карты, многие факс-модемы и т.д. выполнены как электронные платы, т.е. смонтированы на одной плате со своим контроллером;

Остальные устройства используют следующий способ подключения: в системный блок компьютера вставляется электронная плата (контроллер), управляющая работой устройства, а само устройство подсоединяется к этой плате кабелем.

#### **Микропроцессор и сопроцессор.**

Микропроцессор. Самым главным элементом в компьютере, его “мозгом”, является микропроцессор – небольшая (в несколько сантиметров) электронная схема, выполняющая все вычисления и обработку информации. Микропроцессор умеет выполнять сотни различных операций и делает это со скоростью в несколько десятков или даже сотен миллионов операций в секунду. Микропроцессоры в порядке возрастания производительности и цены: Intel-8088, 80286, 80386 (модификации SX и DX), 80486 (модификации SX, SX2, DX, DX2 и DX4), Pentium, Pentium Pro (3,4).

Тактовая частота. Тактовая частота указывает скорость выполнения элементарных операций внутри микропроцессора. Разные модемы микропроцессоров выполняют одни и те же команды (например сложение или умножение) за разное число тактов. Чем более современная модель микропроцессора, тем как правило, меньше тактов требуется микропроцессору для выполнения команд.

Сопроцессор. В тех случаях когда на компьютере приходится выполнять много математических вычислений (например, в инженерных расчетах, обработке трехмерных изображений и т.д.), желательно, чтобы математические операции над вещественными числами поддерживались аппаратно, т.е. самим микропроцессором. Но микропроцессоры не обеспечивают такую поддержку, поэтому к ним для этого требуется добавить математический сопроцессор, который помогает основному микропроцессору выполнять математические операции над вещественными числами. Новейшие микропроцессоры (Pentium) сами умеют выполнять операции над вещественными числами, поэтому для них сопроцессоры не требуются.

#### **Память.**

Оперативная память. Очень важным элементом компьютера является оперативная память. Именно из нее процессор берет программы и исходные данные для обработки, в нее он записывает полученные результаты. Название “оперативная” эта память получила потому, что она работает очень быстро, так что процессору практически не приходится ждать при чтении данных из памяти или записи в память. Однако содержащиеся в ней данные сохраняются только пока компьютер включен. При выключении компьютера содержимое оперативной памяти стирается (за некоторыми исключениями). Часто для оперативной памяти используют обозначение RAM (random access memory, т.е. память с произвольным доступом). От количества установленной в компьютере оперативной памяти напрямую зависит, с какими программами вы сможете на нем работать. При недостаточном количестве оперативной памяти многие программы либо вовсе не будут работать, либо станут работать крайне медленно.

Кэш-память. Для ускорения доступа к оперативной памяти на быстродействующих компьютерах используется специальная сверхбыстродействующая кэш-память, которая располагается как бы между микропроцессором и оперативной памятью и хранит копии наиболее часто используемых участков оперативной памяти. При обращении микропроцессора к памяти сначала производится поиск нужных данных в кэш-памяти. Поскольку время доступа к кэш-памяти в несколько раз меньше, чем к обычной памяти, а в большинстве случаев необходимые микропроцессору данные уже содержатся в кэш-памяти, среднее время доступа к памяти уменьшается.

BIOS (постоянная память). В компьютерах имеется также и постоянная память, в которую данные занесены при ее изготовлении. Как правило, эти данные не могут быть изменены, выполняемые на компьютере программы могут их только считывать. Такой вид памяти обычно называется ROM (read only memory, или память только для чтения), или ПЗУ (постоянное запоминающее устройство). В постоянной памяти хранятся программы для проверки оборудования компьютера, инициализации загрузки ОС и выполнения базовых функций по обслуживанию устройств компьютера. Поскольку большая часть этих программ связана с обслуживанием ввода-вывода, часто содержимое постоянной памяти называется BIOS (basic input-output system, или базовая система ввода-вывода). В BIOS содержится также программа настройки конфигураций компьютера (SETUP). Она позволяет установить некоторые характеристики устройств компьютера (типы видеоконтроллера, жестких дисков и дисководов для дискет, часто также режимы работы с оперативной памятью, запрос пароля при начальной загрузке, и т.д.). Как правило, программа настройки конфигураций вызывается, если пользователь во время начальной загрузки нажмет определенную клавишу или комбинацию клавиш.

CMOS (полупостоянная память). Кроме обычной оперативной памяти и постоянной памяти в компьютере имеются также небольшой участок памяти для хранения параметров и конфигураций компьютера. Его часто называют CMOS-памятью, поскольку эта память выполняется по технологии CMOS (complementary metal-oxide semiconductor), обладающий низким энергопотреблением. Содержимое CMOS-памяти не изменяется при выключении питания компьютера, поскольку для ее питания используется специальный аккумулятор.

Видеопамять. Еще один вид памяти в компьютерах – это видеопамять, т.е. память, используемая для хранения изображения, выводимая на экран монитора. Эта память обычно входит в состав видеоконтроллера – электронной схемы, управляющей выводом изображения на экран.

Электронные платы, контроллеры и шины

Электронные платы. Электронная начинка ПК, как правило, выполняется из нескольких модулей – электронных плат. Каждая плата представляет собой плоский кусок пластика, на котором укреплены электронные компоненты (микросхемы, конденсаторы и т.д.) и различные разъемы. Внутри электронной платы проложены проводники для соединения смонтированных на плате компонент между собой.

Материнская плата. Самой большой электронной платой в компьютере является системная, или материнская, плата. На ней обычно располагаются основной микропроцессор, оперативная память, кэш-память, шины и BIOS. Кроме того, там находятся электронные схемы (контроллеры), управляющие некоторыми устройствами компьютера. Так, контроллер клавиатуры всегда находится на материнской плате. Часто там же находятся и контроллеры для других устройств (жестких дисков, дисководов для дискет и т.д.).

Контроллеры. Электронные схемы, управляющие различными устройствами компьютера, называются контроллерами. Во всех компьютерах имеются контроллеры для управления клавиатурой, монитором, дисководами, жестким диском и т.д.

Интегрированные контроллеры. В современных компьютерах многие контроллеры входят в состав материнской платы. Такие контроллеры называются встроенными или интегрированными. Так, контроллер клавиатуры всегда является встроенным. На современных материнских платах обычно имеются контроллеры дискет, портов ввода-вывода, контроллер жестких дисков, иногда – видеоконтроллер.

Платы контроллеров. Разным пользователям в компьютере нужен разный набор контроллеров. Поэтому все контроллеры компьютера встраиваются в материнскую плату только в некоторых специальных компьютерах. В большинстве компьютеров некоторые контроллеры располагаются на отдельных электронных платах – платах контроллеров. Эти платы вставляются в специальные разъемы (слоты) на материнской плате компьютера.

С помощью добавления и замены плат контроллеров пользователь может модифицировать компьютер, расширяя его возможности и настраивая его по своим потребностям. Например, пользователь может добавить в компьютер факс-модем, звуковую карту, плату приема телепередач и т.д.

Шины. При вставке в разъем материнской платы контроллер подключается к шине – магистрали передачи данных между оперативной памятью и контроллерами. В современных компьютерах обычно имеются две шины:

Шина ISA для контроллеров низкоскоростных устройств (т.е. для обмена данными с клавиатурой, мышью, дисководами, модемом, звуковой картой и т.д.)

Шина PCI для обмена данными с высокоскоростными устройствами (жесткими дисками, видеоконтроллером и т.д.)

В более старых компьютерах могут быть и другие шины – EISA, VESA (VLB) и др. Впрочем, для обеспечения совместимости даже современные серверы локальных сетей обычно оснащаются шиной EISA.

Разъемы шин. Каждый контроллер может быть подключен лишь к той шине, на которую он рассчитан. Поэтому разъемы различных шин сделаны разными, чтобы их нельзя было перепутать. При покупке контроллеров следует знать, разъемы каких шин имеются в вашем компьютере, так как иначе купленный контроллер окажется бесполезен.

Контроллеры портов ввода-вывода. Одним из контроллеров, которые присутствуют почти в каждом компьютере, является контроллер портов ввода-вывода. Часто этот контроллер интегрирован в состав материнской платы. Контроллер портов ввода-вывода соединяется кабелями с разъемами на задней стенке компьютера, через которые к компьютеру подключаются принтер, мышь и некоторые другие устройства. Порты ввода-вывода бывают следующих типов:

Параллельные (обозначаемые LPT1 – LPT4). К соответствующим разъемам на задней стенке компьютера (имеющим 25 гнезд) обычно подключаются принтеры.

Последовательные (обозначаемые COM1 – COM3). К соответствующим разъемам на задней стенке компьютера (имеющим 9 или 25 штырьков) обычно подсоединяются мышь, модем и др. устройства.

Игровой порт. К его разъему (имеющему 15 гнезд) подключается джойстик. Игровой порт имеется не у всех компьютеров.

Как правило, контроллер портов компьютера поддерживает один параллельный и два последовательных порта.

Разъемы шины USB. В компьютерах имеются разъемы универсальной последовательной шины USB. Повидимому скоро будут выпущены модели клавиатур, мышей, принтеров, модемов, дисководов, сканеров и т.д., подключаемые к шине USB. При этом к каждому устройству, подключенному к шине USB, можно подключать другие USB-устройства (всего может быть подключено до 127 устройств). USB-устройство можно подсоединять и отсоединять при работающем компьютере. Возможно, в недалеком будущем в компьютерах вместо разъемов клавиатуры, портов и джойстика будут иметься только два-три маленьких разъема USB.

### **Раздел 3. Технические средства организации информационных процессов**

#### **Тема 3.3. ПЭВМ, ее устройство и основные технические характеристик**

Сеть представляет собой не просто компьютеры, соединенные вместе кабелем. Сеть - это набор компьютеров, осуществляющих обмен данными между собой с определенными целями. Независимо от используемых на каждом компьютере приложений все машины сети делятся на два класса - серверы и рабочие станции.

Сервером будем называть компьютер, предоставляющий свои ресурсы (например, диски) другим компьютерам сети. Серверы предоставляют свои ресурсы рабочим станциям.

Рабочая станция или клиент использует ресурсы сервера. Рабочие станции имеют доступ к сетевым ресурсам, но своих ресурсов в общее пользование не предоставляют.

С сетевыми ресурсами обычно связывают локальные имена (A-Z для дисков; LPTx, COMx - для портов).



Сети с выделенными серверами и одноранговые сети

Сети с архитектурой клиент-сервер используют центральный сервер для обслуживания запросов клиентов, тогда как одноранговые сети позволяют любой рабочей станции функционировать одновременно в качестве сервера, если этого требуют задачи.

По сравнению с универсально одноранговой архитектурой сеть клиент-сервер более специализирована.

Сети с архитектурой клиент-сервер

Специализированный компьютер (выделенный сервер) используется для установки всех разделяемых ресурсов. Такое решение ускоряет доступ пользователей к централизованным ресурсам сети.

Сетевое администрирование проще за счет незначительного числа серверов в сети и их узкой специализации.

Высокие требования к выделенному серверу обеспечение высокой производительности требует установки на сервере большого количества оперативной памяти, диска большого размера и использования в сервере производительного процессора.

При нарушении работы сервера сеть становится практически неработоспособной.

Одноранговые сети

Сетевые приложения могут быть распределены по многочисленным серверам для повышения производительности сети и снижения расходов.

Гибкое разделение ресурсов любого узла сети.

Администрирование одноранговой сети может быть сложнее за счет большего числа серверов и более развитых возможностей каждого сервера.

Невыделенные серверы медленнее специализированных.

В сети LANtastic могут использоваться как выделенные серверы, так и невыделенные серверы/рабочие станции.

Вычислительный процесс в рамках компьютерной сети может быть организован одним из двух способов:

1. Основная нагрузка полагается на рабочие станции, сетевые ресурсы рассматриваются как вспомогательные. В этом случае сеть, как правило, является одноранговой. Рабочие станции представляют собой мощные персональные компьютеры, оснащенные достаточно большим объемом оперативной и внешней памяти, а также устройствами вводу-вывода информации на магнитных носителях.

2. На рабочие станции полагается минимальный объем работы, необходимый только для обеспечения доступа к сетевым ресурсам. Основная нагрузка возлагается на сетевые серверы. В этом случае компьютерная сеть организована по типу "клиент-сервер". Объем оперативной памяти может быть небольшим, а внешняя память и устройства вводу-вывода могут быть вообще отсутствуют. В качестве такой рабочей станции может использоваться специализированный компьютер – сетевая станция. В состав сетевой станции входят такие устройства персонального компьютера: клавиатура, монитор, процессор, видеокарта, системная плата и блок питания. В отличие от персонального компьютера в сетевой станции отсутствуют накопление на жестком магнитном диске (винчестер) и дисководы гибких магнитных дисков – вся необходимая для работы информация сохраняется на сетевом сервере. Отсутствие дисководов для гибких магнитных дисков, кроме снижения стоимости, сетевой станции служит дополнительной защитой от несанкционированного копирования информации и защиты от вирусов. Для связи с локальной сетью в сетевой станции используется встроенный сетевой адаптер (сетевая карта). При отсутствии винчестера инициализация сетевой станции осуществляется с помощью специального запоминающего устройства, размещенного на сетевой карте (BOOTROM).

Кластеры и мэйнфреймы

### Кластерные архитектуры

Двумя основными проблемами построения вычислительных систем для критически важных приложений, связанных с обработкой транзакций, управлением базами данных и обслуживанием телекоммуникаций, являются обеспечение высокой производительности и продолжительного функционирования систем. Наиболее эффективный способ достижения заданного уровня производительности - применение параллельных масштабируемых архитектур.

В последние годы в литературе по вычислительной технике все чаще употребляется термин "системы высокой готовности" (High Availability Systems). Все типы систем высокой готовности имеют общую цель - минимизацию времени простоя. Имеется два типа времени простоя компьютера: плановое и неплановое. Ми-

нимизация каждого из них требует различной стратегии и технологии. Плановое время простоя обычно включает время, принятое руководством, для проведения работ по модернизации системы и для ее обслуживания. Неплановое время простоя является результатом отказа системы или компонента. Хотя системы высокой готовности возможно больше ассоциируются с минимизацией неплановых простоев, они оказываются также полезными для уменьшения планового времени простоя.

Существует несколько типов систем высокой готовности, отличающиеся своими функциональными возможностями и стоимостью. Следует отметить, что высокая готовность не дается бесплатно. Стоимость систем высокой готовности на много превышает стоимость обычных систем. Вероятно поэтому наибольшее распространение в мире получили кластерные системы, благодаря тому, что они обеспечивают достаточно высокий уровень готовности систем при относительно низких затратах. Термин "кластеризация" на сегодня в компьютерной промышленности имеет много различных значений. Строгое определение могло бы звучать так: "реализация объединения машин, представляющегося единым целым для операционной системы, системного программного обеспечения, прикладных программ и пользователей". Машин, кластеризованные вместе таким способом могут при отказе одного процессора очень быстро перераспределить работу на другие процессоры внутри кластера. Это, возможно, наиболее важная задача многих поставщиков систем высокой готовности.

Первой концепцией кластерной системы анонсировала компания DEC, определив ее как группу объединенных между собой вычислительных машин, представляющих собой единый узел обработки информации. По существу VAX-кластер представляет собой слабосвязанную многомашинную систему с общей внешней памятью, обеспечивающую единый механизм управления и администрирования. В настоящее время на смену VAX-кластерам приходят UNIX-кластеры. При этом VAX-кластеры предлагают проверенный набор решений, который устанавливает критерии для оценки подобных систем.

VAX-кластер обладает следующими свойствами:

Разделение ресурсов. Компьютеры VAX в кластере могут разделять доступ к общим ленточным и дисковым накопителям. Все компьютеры VAX в кластере могут обращаться к отдельным файлам данных как к локальным.

Высокая готовность. Если происходит отказ одного из VAX-компьютеров, задания его пользователей автоматически могут быть перенесены на другой компьютер кластера. Если в системе имеется несколько контроллеров внешних накопителей и один из них отказывает, другие контроллеры автоматически подхватывают его работу.

Высокая пропускная способность. Ряд прикладных систем могут пользоваться возможностью параллельного выполнения заданий на нескольких компьютерах кластера.

Удобство обслуживания системы. Общие базы данных могут обслуживаться с единственного места. Прикладные программы могут устанавливаться только однажды на общих дисках кластера и разделяться между всеми компьютерами кластера.

Расширяемость. Увеличение вычислительной мощности кластера достигается подключением к нему дополнительных VAX-компьютеров. Дополнительные накопители на магнитных дисках и магнитных лентах становятся доступными для всех компьютеров, входящих в кластер.

Мейнфреймы

Мейнфрейм - это синоним понятия "большая универсальная ЭВМ". Мейнфреймы и до сегодняшнего дня остаются наиболее мощными (не считая суперкомпьютеров) вычислительными системами общего назначения, обеспечивающими непрерывный круглосуточный режим эксплуатации. Они могут включать один или несколько процессоров, каждый из которых, в свою очередь, может оснащаться векторными сопроцессорами (ускорителями операций с суперкомпьютерной производительностью). В нашем сознании мейнфреймы все еще ассоциируются с большими по габаритам машинами, требующими специально оборудованных помещений с системами водяного охлаждения и кондиционирования. Однако это не совсем так. Прогресс в области элементно-конструкторской базы позволил существенно сократить габариты основных устройств. Наряду со сверхмощными мейнфреймами, требующими организации двухконтурной водяной системы охлаждения, имеются менее мощные модели, для охлаждения которых достаточно принудительной воздушной вентиляции, и модели, построенные по блочно-модульному принципу и не требующие специальных помещений и кондиционеров.

Основными поставщиками мейнфреймов являются известные компьютерные компании IBM, Amdahl, ICL, Siemens Nixdorf и некоторые другие, но ведущая роль принадлежит безусловно компании IBM. Именно архитектура системы IBM/360, выпущенной в 1964 году, и ее последующие поколения стали образцом для подражания. В нашей стране в течение многих лет выпускались машины ряда ЕС ЭВМ, являвшиеся отечественным аналогом этой системы.

В архитектурном плане мейнфреймы представляют собой многопроцессорные системы, содержащие один или несколько центральных и периферийных процессоров с общей памятью, связанных между собой высокоскоростными магистралями передачи данных. При этом основная вычислительная нагрузка ложится на центральные процессоры, а периферийные процессоры (в терминологии IBM - селекторные, блок-мультиплексные, мультиплексные каналы и процессоры телеобработки) обеспечивают работу с широкой номенклатурой периферийных устройств.

Первоначально мейнфреймы ориентировались на централизованную модель вычислений, работали под управлением патентованных операционных систем и имели ограниченные возможности для объединения в единую систему оборудования различных фирм-поставщиков. Однако повышенный интерес потребителей к открытым системам, построенным на базе международных стандартов и позволяющим достаточно эффективно использовать все преимущества такого подхода, заставил поставщиков мейнфреймов существенно расширить

возможности своих операционных систем в направлении совместимости. В настоящее время они демонстрируют свою "открытость", обеспечивая соответствие со спецификациями POSIX 1003.3, возможность использования протоколов межсоединений OSI и TCP/IP или предоставляя возможность работы на своих компьютерах под управлением операционной системы UNIX собственной разработки.

Стремительный рост производительности персональных компьютеров, рабочих станций и серверов создал тенденцию перехода с мейнфреймов на компьютеры менее дорогих классов: миникомпьютеры и многопроцессорные серверы. Эта тенденция получила название "разукрупнение" (downsizing). Однако этот процесс в самое последнее время несколько замедлился. Основной причиной возрождения интереса к мейнфреймам эксперты считают сложность перехода к распределенной архитектуре клиент-сервер, которая оказалась выше, чем предполагалось. Кроме того, многие пользователи считают, что распределенная среда не обладает достаточной надежностью для наиболее ответственных приложений, которой обладают мейнфреймы.

Очевидно выбор центральной машины (сервера) для построения информационной системы предприятия возможен только после глубокого анализа проблем, условий и требований конкретного заказчика и долгосрочного прогнозирования развития этой системы.

Главным недостатком мейнфреймов в настоящее время остается относительно низкое соотношение производительность/стоимость. Однако фирмами-поставщиками мейнфреймов предпринимаются значительные усилия по улучшению этого показателя.

Следует также помнить, что в мире существует огромная инсталлированная база мейнфреймов, на которой работают десятки тысяч прикладных программных систем. Отказаться от годами наработанного программного обеспечения просто не разумно. Поэтому в настоящее время ожидается рост продаж мейнфреймов по крайней мере до конца этого столетия. Эти системы, с одной стороны, позволят модернизировать существующие системы, обеспечив сокращение эксплуатационных расходов, с другой стороны, создадут новую базу для наиболее ответственных приложений.

## **Раздел 4. Программные средства реализации информационных процессов**

### **Тема 3.1. Операционные системы**

Операционная система (ОС) - комплекс системных и управляющих программ, предназначенных для наиболее эффективного использования всех ресурсов вычислительной системы (ВС) (Вычислительная система - взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации) и удобства работы с ней.

В программном обеспечении операционная система занимает основное положение, поскольку осуществляет планирование и контроль всего вычислительного процесса. Любая из компонент программного обеспечения обязательно работает под управлением ОС.

Назначение ОС - организация вычислительного процесса в вычислительной системе, рациональное распределение вычислительных ресурсов между отдельными решаемыми задачами; предоставление пользователям многочисленных сервисных средств, облегчающих процесс программирования и отладки задач. Операционная система исполняет роль своеобразного интерфейса (Интерфейс - совокупность аппаратуры и программных средств, необходимых для подключения периферийных устройств к ПЭВМ) между пользователем и ВС, т.е. ОС предоставляет пользователю виртуальную ВС. Это означает, что ОС в значительной степени формирует у пользователя представление о возможностях ВС, удобстве работы с ней, ее пропускной способности. Различные ОС на одних и тех же технических средствах могут предоставить пользователю различные возможности для организации вычислительного процесса или автоматизированной обработки данных. В соответствии с условиями применения различают три режима ОС: пакетной обработки, разделения времени и реального времени. В режиме пакетной обработки ОС последовательно выполняет собранные в пакет задания. В этом режиме пользователь не имеет контакта с ЭВМ, получая лишь результаты вычислений. В режиме разделения времени ОС одновременно выполняет несколько задач, допуская обращение каждого пользователя к ЭВМ. В режиме реального времени ОС обеспечивает управление объектами в соответствии с принимаемыми входными сигналами. Время отклика ЭВМ с ОС реального времени на возмущающее воздействие должно быть минимальным.

Операционная система в наибольшей степени определяет облик всей вычислительной системы в целом. Несмотря на это, пользователи, активно использующие вычислительную технику, зачастую испытывают затруднения при попытке дать определение операционной системе. Частично это связано с тем, что ОС выполняет две по существу мало связанные функции: обеспечение пользователю-программисту удобств посредством предоставления для него расширенной машины и повышение эффективности использования компьютера путем рационального управления его ресурсами.

Использование большинства компьютеров на уровне машинного языка затруднительно, особенно это касается ввода-вывода. Например, для организации чтения блока данных с гибкого диска программист может использовать 16 различных команд, каждая из которых требует 13 параметров, таких как номер блока на диске, номер сектора на дорожке и т. п. Когда выполнение операции с диском завершается, контроллер возвращает 23 значения, отражающих наличие и типы ошибок, которые, очевидно, надо анализировать. Даже если не входить в курс реальных проблем программирования ввода-вывода, ясно, что среди программистов нашлось бы не много желающих непосредственно заниматься программированием этих операций. При работе с диском программисту-пользователю достаточно представлять его в виде некоторого набора файлов, каждый из которых имеет имя. Работа с файлом заключается в его открытии, выполнении чтения или записи, а затем в закрытии файла. Вопросы подобные таким, как следует ли при записи использовать усовершенствованную частотную модуляцию или в каком состоянии сейчас находится двигатель механизма перемещения считывающих головок, не должны волновать пользователя. Программа, которая скрывает от программиста все реалии аппаратуры и

предоставляет возможность простого, удобного просмотра указанных файлов, чтения или записи - это, конечно, операционная система. Точно также, как ОС ограждает программистов от аппаратуры дискового накопителя и предоставляет ему простой файловый интерфейс, операционная система берет на себя все малоприятные дела, связанные с обработкой прерываний, управлением таймерами и оперативной памятью, а также другие низкоуровневые проблемы. В каждом случае та абстрактная, воображаемая машина, с которой, благодаря операционной системе, теперь может иметь дело пользователь, гораздо проще и удобнее в обращении, чем реальная аппаратура, лежащая в основе этой абстрактной машины.

С этой точки зрения функцией ОС является предоставление пользователю некоторой расширенной или виртуальной машины, которую легче программировать и с которой легче работать, чем непосредственно с аппаратурой, составляющей реальную машину.

Идея о том, что ОС прежде всего система, обеспечивающая удобный интерфейс пользователям, соответствует рассмотрению сверху вниз. Другой взгляд, снизу вверх, дает представление об ОС как о некотором механизме, управляющем всеми частями сложной системы. Современные вычислительные системы состоят из процессоров, памяти, таймеров, дисков, накопителей на магнитных лентах, сетевых коммуникационной аппаратуры, принтеров и других устройств. В соответствии со вторым подходом функцией ОС является распределение процессоров, памяти, устройств и данных между процессами, конкурирующими за эти ресурсы. ОС должна управлять всеми ресурсами вычислительной машины таким образом, чтобы обеспечить максимальную эффективность ее функционирования. Критерием эффективности может быть, например, пропускная способность или реактивность системы. Управление ресурсами включает решение двух общих, не зависящих от типа ресурса задач:

планирование ресурса - то есть определение, кому, когда, а для делимых ресурсов и в каком количестве, необходимо выделить данный ресурс;

отслеживание состояния ресурса - то есть поддержание оперативной информации о том, занят или не занят ресурс, а для делимых ресурсов - какое количество ресурса уже распределено, а какое свободно.

Для решения этих общих задач управления ресурсами разные ОС используют различные алгоритмы, что в конечном счете и определяет их облик в целом, включая характеристики производительности, область применения и даже пользовательский интерфейс. Так, например, алгоритм управления процессором в значительной степени определяет, является ли ОС системой разделения времени, системой пакетной обработки или системой реального времени.

## **Раздел 4. Программные средства реализации информационных процессов**

### **Тема 3.2. Сервисное программное обеспечение**

Сервисное программное обеспечение – это совокупность программных продуктов, предоставляющих пользователю дополнительные услуги в работе с компьютером и расширяющих возможности операционных систем. По функциональным возможностям сервисные средства можно подразделять на:

- защищающие данные от разрушения и несанкционированного доступа;
- восстанавливающие данные;
- ускоряющие обмен данными между диском и ОЗУ;
- архивации-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 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 файлов. Среди имеющихся оболочек для семейства MS-DOS наиболее популярна оболочка Norton Commander.

Утилиты – программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров.

Наиболее часто используются утилиты следующего назначения:

Программы резервирования – создают резервные копии информации на дисках.

Антивирусные программы – предназначены для предотвращения заражения компьютерным вирусом и ликвидации последствий заражения.

Программы-упаковки позволяют за счет применения специальных методов «упаковки», сжимать информацию на дисках, а также объединять копии нескольких файлов в один архивный файл.

Программы-русификаторы, приспособливают другую программу для работы с русскими буквами и текстами (а иногда переводят на русский язык её меню и сообщения).

Программы удаления приложений. Многие комплексы программ содержат программы установки, но не содержат средств, для удаления этих комплексов. Для корректного удаления комплексов программ используются программы удаления приложений.

Программы для оптимизации дисков позволяют обеспечить более быстрый доступ к информации на диске за счет оптимизации размещения данных на диске. Эти программы перемещают все участки каждого файла

друг к другу, собирают все файлы в начале диска и т. д. Из программ для оптимизации широко используется SpeeDisk.

Программы ограничения доступа к данным. Во многих случаях нужно защитить компьютер от нежелательных пользователей. Программа Norton DiskLock защищает паролем компьютер, не позволяя загрузить компьютер посторонним, запрашивает пароль после перерыва в работе и т. д.

Программы для управления памятью обеспечивают более гибкое использование оперативной памяти компьютера.

Программы-кэши для диска убыстряют доступ к информации на диске путем организации в оперативной памяти кэш-буфера, содержащего наиболее часто используемые участки диска. Утилиты предоставляют пользователю дополнительные услуги (не требующие разработки специальных программ) в основном по обслуживанию дисков и файловой системы.

#### **Работа с дисками**

Есть ряд служебных программ для работы с дисками. Они выполняют проверку на наличие ошибок, дефрагментацию, архивацию, а также очистку диска. Все эти программы можно запустить из главного меню Windows.

Проверка диска. Вместе с Windows поставляется утилита Scandisk, с помощью которой пользователь может искать и исправлять ошибки и сбои на дискетах и жестких дисках. Обслуживание дисков с её помощью должно стать неотъемлемой частью профилактической работы.

Во время стандартного теста Scandisk проверяет файлы и папки на логические ошибки и, если Вы это указали, автоматически исправляет их. Scandisk ищет пересекающиеся (cross-linked) файлы, у которых информация оказалась хранящейся в одних и тех же участках диска. Информация, хранящаяся в общих участках, может принадлежать только одному файлу. Scandisk также ищет "потерянные" кластеры - фрагменты диска, потерявшие связь со своими файлами, но по-прежнему считающиеся занятыми. Хотя они и могут содержать полезную информацию, ее, как правило, невозможно восстановить, и она только зря занимает место на диске.

Диск можно проверить не только на логические, но и на физические ошибки, к которым относятся повреждения магнитного покрытия диска. Если Scandisk обнаружит поврежденную область, то он переместит всю оставшуюся информацию в неповрежденное место диска, а данная область будет помечена как сбойная, и информация больше никогда не будет туда записываться.

При неправильном выключении компьютера часто возникают потерянные участки на диске. В связи с этим при следующем включении программа тестирования состояния диска запускается автоматически. Среди утилит, аналогичных Scandisk, можно выделить программу Norton Disk Doctor, входящую в состав программного пакета Norton Utilities.

Профилактическое тестирование дисков необходимо выполнять регулярно. Для жесткого диска можно рекомендовать стандартную проверку не реже одного раза в 1-2 недели. Так как поверхность жесткого диска обычно изнашивается медленно, то полную проверку достаточно выполнять один раз в месяц. Состояние поверхности дискет обязательно нужно проверять после форматирования, так как при этом многие сбойные места остаются не выявленными.

#### **Очистка диска**

При работе с компьютером постепенно на дисках накапливается некоторое количество ненужных файлов. Многие программы создают временные файлы, но не все удаляют их по окончании работы. Конечно, о большинстве файлов сказать, нужны ли они или нет, можете только вы, однако при своей работе, система Windows сама создает множество файлов, которые занимают место на диске впустую.

Во время работы Windows на диске образуется много файлов, которые создаются для временных целей. Когда на диске много места, количество этих файлов не имеет большого значения. Когда место на диске заканчивается, удаление лишних файлов может помочь в решении проблемы. При просмотре диска подлежащими удалению рассматриваются несколько групп файлов. Прежде всего, это содержимое папки Корзина (Recycle Bin), в которую попадают удаляемые вами файлы, поскольку большинство файлов, попадающих в нее, предназначено для окончательного уничтожения. Далее идет папка Temporary Internet Files (Временные файлы Интернета), содержащая файлы, которые появились на компьютере при работе с Интернетом. Эти файлы нужны для быстрого повторного просмотра Web-страниц, на которых вы уже побывали, и при их удалении ничего страшного не произойдет.

#### **Дефрагментация диска**

Система хранения данных на жестком диске в системе Windows построена так, что постепенно работа с диском может несколько замедляться. Все дело в принципах работы файловых систем FAT, FAT32 и NTFS, которые используются Windows. В этих файловых системах весь диск делится на мелкие части одинакового размера, называемые кластерами. При создании нового файла система Windows помещает в специальной области в начале диска запись, где содержится имя файла и номер первого кластера, куда будет записываться файл. Если файл большой, и одного кластера не хватило, то система ищет первый попавшийся свободный кластер и пишет в него остаток файла. Так продолжается до тех пор, пока весь файл не будет записан на диск. Последний кластер файла помечается особо. Все кластеры, использованные в записи файла, помечаются как занятые. При удалении файла все кластеры помечаются как свободные.

Такая система хорошо работает в самом начале, когда свободные кластеры расположены по порядку. Однако в процессе работы, во время создания и удаления файлов свободные кластеры могут появляться в произвольном месте диска, и в скором времени файлы на диске становятся фрагментированными. В таком файле часть информации может находиться в начале диска, а часть в конце. Чтение и запись такого файла существен-

но замедляется, так как диску приходится постоянно перемещать головки из одного места в другое, а это занимает время. Особенно заметно уменьшение скорости работы при фрагментации дисков с файловыми системами FAT и FAT32. Благодаря особенностям построения файловой системы NTFS, уменьшение скорости из-за эффекта фрагментации незначительно, хотя и система NTFS подвержена фрагментации. Чтобы принудительно ликвидировать фрагментацию, то есть выполнить дефрагментацию диска, в Windows предусмотрена специальная программа, входящая в состав стандартных служебных программ.

Перед дефрагментацией желательно протестировать диск программой Scandisk, чтобы удостовериться, что запись на диск будет происходить нормально. Обращаем ваше внимание, что перед запуском программы дефрагментации необходимо завершить работу всех остальных программ. Это связано с тем, что любое изменение информации на обрабатываемом диске, а это может случиться, если другие программы работают, приводит к повторному запуску дефрагментации.

## **Раздел 4. Программные средства реализации информационных процессов**

### **Тема 3.3. Офисное программное обеспечение**

Компьютеры чаще всего используются для выполнения следующих работ:

- обработки входящей и исходящей информации с помощью текстовых редакторов и средств презентационной графики;
- сбора и анализа данных, расчетов и отчетов, выполняемых обычно с использованием электронных таблиц;
- накопления и хранения поступившей информации, обеспечивающих быстрый ее поиск.

В пакет офисного программного обеспечения Microsoft входят:

Office Базовый Office Стандартный Office для дома и учебы Office для малого бизнеса Office Профессиональный Office Профессиональный плюс Office Корпоративный

В базовый пакет офисного программного обеспечения входят:

Microsoft Office Access  
Microsoft Office Excel  
Microsoft Office FrontPage  
Microsoft Office Outlook  
Microsoft Office PowerPoint  
Microsoft Office Publisher  
Microsoft Office Word  
Microsoft Office OneNote  
Microsoft Office Publisher  
Microsoft Office InfoPath  
Microsoft Office Communicator  
Microsoft Office Groove

Эти программы позволяют автоматизировать большинство процессов офиса, от составления простейших баз данных до редактирования и оформления текстовой документации. Пакет программ Microsoft Office является незаменимым и универсальным инструментом работы современной организации.

Microsoft Word — один из лучших текстовых редакторов. Он позволяет создавать и редактировать документы, добавлять в них таблицы и рисунки, изменять оформление абзацев и начертание шрифта, готовить документ к печати. Дополнительные модули Word позволяют выполнять такие операции, как проверка орфографии и грамматики, формирование оглавлений и указателей, слияние с базой данных. Microsoft Word — мощный текстовый процессор. С его помощью можно не только отформатировать и красиво оформить многостраничный текст, но даже полностью сверстать небольшую книгу.

Окно Word обладает всеми стандартными элементами текстового процессора. В заголовок выводится название документа. Строка меню обеспечивает доступ к командам. Панели инструментов содержат кнопки самых популярных операций. Строка состояния сообщает о количестве страниц и разделов в документе, обозначает текущее положение текстового курсора и содержит индикаторы некоторых режимов программы. Вертикальная полоса прокрутки обеспечивает быстрое перемещение по тексту документа вверх и вниз. Здесь есть линейка, облегчающая выравнивание объектов, и кнопки смены режимов просмотра документа.

Word является многофункциональной программой, существенно сокращающей затраты на создание и оформление документов и позволяющей:

- вводить текст с одновременным контролем орфографии и даже стилистики на экране;
- легко вносить исправления и изменения в тексте;
- оформлять документ в соответствии с его назначением;
- сохранить полученный результат для последующего использования;
- добавлять в текст таблицы, графические объекты и т. д.;
- использовать готовые шаблоны документов;
- печатать документ на различных принтерах;
- автоматически создавать указатели и оглавления;
- отправлять документы по факсу или по электронной почте.

Microsoft Excel — программа, обладающая эффективными средствами обработки числовой информации, представленной в виде электронных таблиц. Она позволяет выполнять математические, финансовые и статистические вычисления, оформлять отчеты, построенные на базе таблиц, выводить числовую информацию в виде графиков и диаграмм.

Документ приложения Excel называется рабочей книгой (workbook) или просто книгой Excel. Такая книга состоит из листов (worksheet), которые представляют собой большие таблицы ячеек с числами и текстовой информацией. Таблицы Excel похожи на базы данных, но предназначены не столько для хранения информации, сколько для проведения математических и статистических расчетов.

#### **Раздел 4. Программные средства реализации информационных процессов**

##### **Тема 3.4. Программное обеспечение для работы с компьютерной графикой**

Программное обеспечение для работы с растровой графикой можно разделить на следующие группы:

Средства создания изображений. Ряд графических редакторов, например, Painter и Fauve Matisse, ориентирован непосредственно на процесс рисования. В них акцент сделан на использование удобных инструментов рисования и на создание новых художественных инструментов и материалов. К простейшим программам этого класса относится также графический редактор Paint.

Средства обработки изображений. Другой класс растровых графических редакторов предназначен не для создания изображений "с нуля", а для обработки готовых рисунков с целью улучшения их качества и реализации творческих идей. К таким программам, в частности, относятся Adobe Photoshop, Photostyler, Picture Publisher и др.

Исходная информация для обработки на компьютере может быть получена разными путями: сканированием цветной иллюстрации, загрузкой изображения, созданного в другом редакторе, или вводом изображения от цифровой фото- или видеокамеры. При создании художественных композиций отдельные фрагменты часто заимствуют из библиотек изображений-клипартов, распространяемых на компакт-дисках. Основа будущего рисунка или его отдельные элементы могут быть созданы и в векторном графическом редакторе, после чего их экспортируют в растровом формате.

3. Средства каталогизации изображений. Особый класс программ для работы с растровыми изображениями представляют программы-каталогизаторы. Они позволяют просматривать графические файлы множества различных форматов, создавать на жестком диске удобные альбомы, перемещать и переименовывать файлы, документировать и комментировать иллюстрации. Удобной программой этого класса считается программа ACDSec.

##### **Понятие векторной и растровой графики**

Одним из популярных направлений использования персонального компьютера является компьютерная графика, которая изучает средства и способы создания, а также обработки изображения с помощью программно-аппаратных вычислительных комплексов.

Компьютерная графика подразделяется на растровую, векторную и фрактальную из-за способа формирования изображений на экране монитора.

##### **Растровая графика**

Растровая графика – это вид компьютерной графики, в котором основным элементом является точка, экранное изображение которой называется пикселем. Растр – это матрица ячеек (пикселей), размер раstra измеряется количеством пикселей по горизонтали и вертикали.

##### **Достоинства:**

Каждый пиксел независим друг от друга

Автоматизации ввода (оцифровки) информации, которая может вводиться с внешних устройств: сканеров, видеокамер, цифровых фотокамер, графических планшетов.

Фотореалистичность – возможность корректировки деталей;

Стандартные форматы файлов, предназначенные для сохранения точечных изображений.

##### **Недостатки**

Большой объем файла;

Искажения при любых трансформациях;

Пикселизация при увеличении точек раstra, что визуально искажает иллюстрацию.

Растровую графику применяют при разработке мультимедийных учебных материалов и полиграфических изданий, создании коллажей и реставрации изображений, восстановлении потерянной информации. Иллюстрации, выполненные средствами растровой графики, редко создают вручную с помощью компьютерных программ. Чаще для этой цели используют сканированные иллюстрации, подготовленные художником на бумаге, или фотографии.

##### **Векторная графика**

Основным элементом в векторной графике является линия. Как известно из математики любую линию можно описать с помощью точек (или узлов в векторной графике), каждая из которых имеет координаты и направление, изменяя эти параметры можно изменить форму линии или ее свойства: форму, толщину, цвет, характер (сплошная, пунктирная и т.п.). Простейшая линия, если она не замкнута, имеет две вершины, которые называются узлами и тоже имеют свойства, от которых зависит, как выглядит вершина линии и как две линии пересекаются между собой. Если линии замкнуты, то они образуют замкнутый контур, имеющий свойство заполнения. Внутренняя область может быть заполнена цветом, текстурой, картой.

Программные средства для работы с векторной графикой предназначены для создания иллюстраций, чертежей, верстки рекламы и др., поэтому часто находят свое применение в рекламных агентствах, дизайнерских бюро, издательствах.

#### **Графические редакторы**

В обширном классе программ для обработки растровой графики особое место занимает пакет Photoshop компании Adobe.

Adobe PhotoShop – широко распространенная программа для обработки растровой графики. Она используется для создания фотореалистических изображений, работы с цветными сканированными изображениями, ретуширования, цветокоррекции, коллажирования, трансформации и цветоделения и др. Adobe PhotoShop располагает всеми традиционными методами работы с точечными изображениями, при этом имеет возможность работы со слоями и использует контуры. Программа позволяет легко изменять цветовое представление документов (битовое, в градациях серого, дуплекс, индексированные цвета, RGB или CMYK). В пакет AdobePhotoshop также входит особая группа программных средств обработки изображений - Фильтры. Это подключаемые к программе модули, часто третьих фирм, позволяющие обрабатывать изображение по заданному алгоритму. Иногда такие алгоритмы бывают очень сложными, а окно фильтра может иметь множество настраиваемых параметров. Из групп фильтров популярны продукты серий Kai's Power Tools, Alien Skin, Andromeda и другие.

К программным средствам создания и обработки векторной графики относятся графические редакторы (например Adobe Illustrator, Macromedia Freehand, CorelDraw) и векторизаторы (трассировщики) - специализированные пакеты преобразования растровых изображений в векторные (например Adobe Streamline, CorelTrace).

Если основным требованием к изображению является высокая точность формы, то применяют специальные графические редакторы, предназначенные для работы с векторной графикой. Такая задача возникает при разработке логотипов компаний, при художественном оформлении текста (например, журнальных заголовков или рекламных объявлений), а также во всех случаях, когда иллюстрация является чертежом, схемой или диаграммой, а не рисунком.

Adobe Illustrator является общепризнанным мировым лидером среди средств векторной графики. Ее особое достоинство состоит в том, что вместе с программами Adobe Photoshop и PageMaker она образует законченное трио приложений, достаточных для выполнения компьютерной верстки полиграфических изданий и разработки сложных документов. Эти приложения выполнены в едином стиле, используют похожие интерфейсы и инструменты, позволяют применять одинаковые приемы и навыки и безошибочно экспортируют и импортируют созданные объекты между собой. Дополнительным преимуществом Adobe Illustrator является тот факт, что этот векторный редактор имеет версию на русском языке. Начиная с 7-ой версии, в программу встроена поддержка цветоделения. В программе более 50 фильтров, которые позволяют создать спецэффекты и имитировать различные манеры художников, а также используются свободно komponуемые палитры. Кроме того, палитру инструментов можно расширить за счет инструментов, созданных другими фирмами. Возможны импорт и экспорт файлов в различных форматах (BMP, TIF и т.д.), возможен импорт и экспорт текстовых файлов, включая такие форматы как ASCII, RTF, Microsoft Word.

Программа обладает мощными средствами для печатной обработки. Можно создать вертикальный текст, по контуру, изменять масштаб и ориентацию отдельных символов, производить верстку текста. Возможно редактировать цвета. Существует до 200 уровней "undo", причем реальное число ограничено только объемом памяти.

Macromedia Freehand 8.0 - очень удобный векторный редактор с простым и дружелюбным интерфейсом служит удобным инструментом работы для начинающих. Программа отличается небольшим размером и хорошим быстродействием. Нетребовательность к аппаратным ресурсам позволяет работать на компьютерах среднего уровня. Несмотря на простоту системы управления, инструментальные средства Macromedia Freehand достаточны для разработки весьма сложных документов и лишь на высоком уровне сложности уступают средствам Adobe Illustrator и CorelDraw. Программу Macromedia Freehand удобно использовать при работе с любыми системами компьютерной верстки, но специально она адаптирована для системы QuarkXPress.

CorelDraw (версии от 5.0 до 11.0). Векторный редактор CorelDraw исторически, считается основным пакетом создания и обработки векторной графики на платформе Windows. Особенно удобен CorelDRAW при создании иллюстраций, состоящих их множества рисунков, фотографий и надписей. В пакет программ CorelDRAW кроме собственно редактора векторной графики входит редактор растровой графики Corel PhotoPaint. Интерфейс обеих программ очень похож, и по возможностям они дополняют друг друга. Используя две эти программы, возможно выполнить практически любую, самую сложную, графическую работу. Также новая программа, входящая в состав пакета CorelDRAW, называемая Corel R.A.V.E., предназначена для создания анимации и мультфильмов на основе векторной графики. Программа Corel R.A.V.E. позволяет экспортировать результат в разнообразные популярные форматы, в том числе в формат Macromedia Flash. В стандартную поставку CorelDRAW входит множество готовых картинок, которые можно использовать при создании собственных иллюстраций. Панели инструментов и горячие клавиши можно перенастроить, при этом пакет позволяет сохранять несколько конфигураций рабочего стола. К преимуществам этого редактора относят развитую систему управления и богатство средств настройки инструментов. В CorelDraw система управления сложнее, чем в других векторных редакторах, и интерфейс не столь интуитивен. Изучение CorelDraw представляет более сложную задачу, чем изучение Adobe Illustrator или Macromedia Freehand.

Трассировщик Adobe Streamline по праву занимает ведущее место в своем классе программ. Хотя имеются более мощные пакеты, ориентированные на обработку чертежей, они очень требовательны к аппаратным ресурсам, да и по стоимости много дороже. Streamline позволяет проводить тонкую настройку параметров векторизации, что улучшает ее точность. Более всего векторизация удобна для преобразования чертежей, черно-

белых рисунков и другой простой графики без полутонов. Полутоновые и цветные изображения обрабатываются хуже, и результат требует значительной доработки для приближения к оригиналу.

## **Раздел 4. Программные средства реализации информационных процессов**

### **Тема 3.5. Системы управления базами данных**

В истории развития вычислительной техники наблюдалось два основных направления её применения. Первое связано с выполнением численных расчетов. Развитие этой области способствовало ускорению развития методом математического моделирования, численных методов, языков программирования высокого уровня. Второе направление связано с использованием вычислительной техники для создания, хранения и обработки больших массивов данных. Такие задачи решают базы данных и системы их управления.

База данных (БД) – это большой массив данных, которые организованы в табличные структуры. Иначе: совокупность взаимосвязанных данных, при такой минимальной избыточности, которая позволяет её использовать оптимальным образом для одного или нескольких приложений в определённой предметной области человеческой деятельности (организованная структура, предназначенная для хранения информации).

База данных содержит методы и средства, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят редактирование.

Явления реального мира зачастую могут быть описаны с помощью структурных взаимосвязей между совокупностями фактов. Для представления информации о таких явлениях может быть использована структурная модель данных. В общем случае выделяют два типа связей данных:

- связь между атрибутами одного и того же объекта;
- связь между объектами.

Связь атрибутов представляется типом связей, которые в свою очередь являются поименованной совокупностью элементов данных. Связи между объектами могут быть представлены некоторым графом или диаграммой структуры данных.

Создание, взаимодействие и поддержание БД в актуальном состоянии осуществляется по средствам системы управления базой данных (СУБД) – комплекса программных сред, предназначенных для создания структуры новой базы, наполнения её, редактирования и визуализации информации (например, MS Access, MS SQL, FoxPro).

Основными функциями систем управления базами данных являются: хранение и ведение представления структурной информации; предоставление средств заполнения базы данных или импорта данных из другой таблицы; обеспечение возможности доступа к данным, а также представление средств поиска и фильтрации.

Многие СУБД дополнительно предоставляют возможности проведения простого анализа данных и их обработку. В результате возможно создание новых таблиц баз данных на основе имеющихся. Сегодня СУБД умеют работать с удаленными и распределенными ресурсами (в сети).

Использование БД обеспечивает выполнение ряда задач:

- независимость данных и программ;
- реализацию отношений между данными;
- совместимость компонентов БД;
- простоту изменения логической и физической структур БД;
- целостность;
- сокращение избыточности в хранимых данных;
- уменьшение стоимости разработки пакета программ;
- программирование запросов к базе данных.

Простейший пример неэлектронной базы данных – это деловой ежедневник.

База данных является динамической информационной моделью предметной области, отображением внешнего мира. Каждому объекту присущ ряд характерных для него свойств, признаков, параметров. Работа с БД осуществляется по атрибутам объектов.

Аналогом столбцов и строк в базе данных выступают соответственно поля и записи. Типы данных, с которыми может работать база данных: текстовые, числовые, логические, дата/время, денежный, счетчик-нумерация.

Обычно с базой данных работают две категории исполнителей: проектировщики и пользователи. Проектировщики занимаются структурой базы данных, пользователи активно её используют и вносят изменения в хранимые данные.

В качестве объектов выступают таблицы и запросы, которые ещё иногда называют моментальным снимком. Все записи в основные таблицы баз данных вносятся по мере их поступления - без сортировки и упорядоченности - в этом ещё одно отличие баз данных от таблиц. Если же пользователю нужно видеть те или иные данные-то он формирует запрос. Средством ввода данных в БД служат формы. Смысл состоит в том, чтобы предоставить пользователю средства для заполнения только тех полей, которые ему положено заполнять по структуре базы данных. Для вывода необходимой информации на печать используются отчеты.

## **Раздел 5. Алгоритмизация и программирование**

### **Тема 5.1. Понятие алгоритма**

Алгоритм - понятное и точное предписание исполнителю совершить последовательность действий (набор операций и правил их чередования), направленных на достижение указанной цели или на решение по-

ставленной задачи. Алгоритм используется в задачах программирования. Как самостоятельная дисциплина, программирование, описывает понятия метода, приёма, средства процессов конструирования и исполнения программ. Обычно, под конструированием в информатике понимают знание и применение приёмов программирования. Результатом решения задачи в программировании является программа – описанием алгоритма обработки информации на определённом языке.

Перед решением любой задачи с помощью персонального компьютера (ПК) выполняются следующие этапы: постановка этой задачи, построение сценария и алгоритмизация. Под алгоритмизацией задачи понимают процесс разработки (проектирования) алгоритма решения задачи с помощью ПК на основе ее условия и требований к конечному результату. На этапе постановки задачи описываются исходные данные и предпосылки, формируются правила начала и окончания решения задачи (достижения цели), т. е. разрабатывается информационная или эквивалентная ей математическая модель.

В общем случае алгоритмизация вычислительного процесса включает следующие действия:

последовательную декомпозицию задачи, выделение автономных этапов вычислительного процесса и разбивку каждого этапа на отдельные шаги;

формальную запись содержания каждого этапа и/или шага;

определение общего порядка выполнения этапов и/или шагов;

проверку правильности алгоритма.

Свойства алгоритма

Понятность для исполнителя - содержание предписания о выполнении только таких действий, которые входят в систему команд исполнителя.

Дискретность (прерывность, раздельность) - выполнение команд алгоритма последовательно, с точной фиксацией моментов окончания выполнения одной команды и начала выполнения следующей.

Определенность - каждое правило алгоритма должно быть четким, однозначным. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

Результативность - либо завершение решения задачи после выполнения алгоритма, либо вывод о невозможности продолжения решения по какой-либо из причин.

Массовость - означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными из некоторой области применимости алгоритма.

Для решения одной и той же задачи, как правило, можно использовать различные алгоритмы. В связи с этим, возникает необходимость сравнивать их между собой, и для этого нужны определенные критерии качества алгоритмов. К ним относят временные и объемные характеристики.

Временные характеристики алгоритма определяют длительность решения или временную сложность. Длительность решения часто выражается в единицах времени, но удобнее ее выражать через количество операций, так как последнее не зависит от быстродействия конкретной машины. Временной сложностью алгоритма называется зависимость времени счета, затрачиваемого на получение результатов от объема исходных данных.

Объемные характеристики алгоритма определяют его информационную сложность, которая связана со сложностью описания, накопления и хранения исходных, промежуточных и результирующих данных при решении определенной задачи. Объем текста алгоритма (программы) определяется количеством операторов, использованных для записи алгоритма.

#### **Способы описания алгоритмов**

Для строгого задания различных структур данных и алгоритмов их обработки требуется иметь такую систему формальных обозначений и правил, чтобы смысл всякого используемого предписания трактовался точно и однозначно. Соответствующие системы правил называются языками описаний. К средствам описания алгоритмов относятся следующие основные способы их представления: словесный, формульно-математический, графический, программный. На практике используются также и табличный способ.

Словесный способ записи алгоритмов представляет собой последовательное описание основных этапов обработки данных и задается в произвольном изложении на естественном языке.

При словесно-математическом способе кроме слов используются математические соотношения.

Формульно-математический способ наиболее близок к математке. Он предполагает использование только формул.

Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным. При графическом представлении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Такое графическое представление называется схемой алгоритма или блок-схемой. В блок-схеме каждому типу действий (вводу исходных данных, вычислению значений выражений, проверке условий, управлению повторением действий, окончанию обработки и т. п.) соответствует геометрическая фигура, представленная в виде блочного символа. Блочные символы соединяются линиями переходов, определяющими очередность выполнения действий. Для начертания этих схем используется набор символов, определяемых ГОСТ 19.701-90 (ИСО 5807 - 85) "Единая система программной документации". Программный способ предполагает использования для описания алгоритма определённый набор команд какого-либо языка программирования. Примером такого способа является программа.

#### **Структурные схемы алгоритмов**

Преобразования величин, реализуемые в алгоритмическом языке, осуществляются по операторам (командам), располагаемым в заданной последовательности. Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: следование, ветвление, цикл.

Структура алгоритма является линейной, если она образована последовательностью простых операторов (команд).

Разветвляющийся алгоритм – алгоритм, содержащий хотя бы одно условие, в результате проверки которого обеспечивается переход на один из двух возможных шагов.

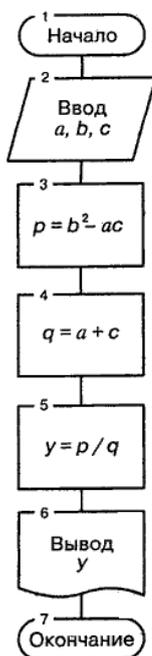
Циклический алгоритм – алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. Группа команд (операторов), выполняющихся одна за другой, называется серией, которая может состоять из одного оператора.

Сложные структуры алгоритма являются совокупностью рассмотренных выше.

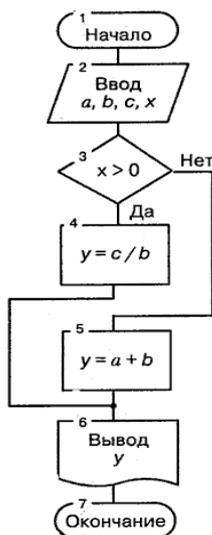
Линейным принято называть вычислительный процесс, в котором операции выполняются последовательно, в порядке их записи. Каждая операция является самостоятельной, независимой от каких-либо условий. На схеме блоки, отображающие эти операции, располагаются в линейной последовательности.

Линейные вычислительные процессы имеют место, например, при вычислении арифметических выражений, когда имеются конкретные числовые данные и над ними выполняются соответствующие условию задачи действия. На рис. показан пример линейного алгоритма, определяющего процесс вычисления арифметического выражения

$$y = (b^2 - ac) : (a + c).$$



Вычислительный процесс называется ветвящимся, если для его реализации предусмотрено несколько направлений (ветвей). Каждое отдельное направление процесса обработки данных является отдельной ветвью вычислений. Ветвление в программе — это выбор одной из нескольких последовательностей команд при выполнении программы. Выбор направления зависит от заранее определенного признака, который может относиться к исходным данным, к промежуточным или конечным результатам. Признак характеризует свойство данных и имеет два или более значений.



Ветвящийся процесс, включающий в себя две ветви, называется простым, более двух ветвей — сложным. Сложный ветвящийся процесс можно представить с помощью простых ветвящихся процессов.

Направление ветвления выбирается логической проверкой, в результате которой возможны два ответа: «да» — условие выполнено и «нет» — условие не выполнено.

Следует иметь в виду, что, хотя на схеме алгоритма должны быть показаны все возможные направления вычислений в зависимости от выполнения определенного условия (или условий), при однократном прохождении программы процесс реализуется только по одной ветви, а остальные исключаются. Любая ветвь, по которой осуществляются вычисления, должна приводить к завершению вычислительного процесса.

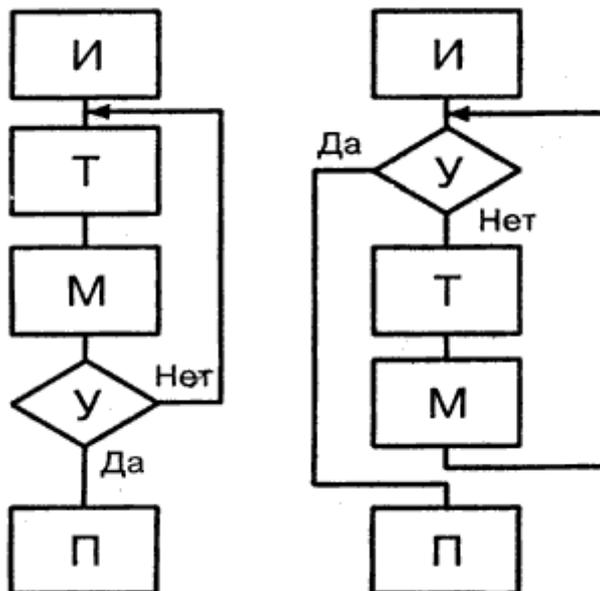
На рис. показан пример алгоритма с разветвлением для вычисления следующего выражения:  $Y = (a+b)$ , если  $X < 0$ ;  $c/b$ , если  $X > 0$ .

Циклическими называются программы, содержащие циклы. Цикл — это многократно повторяемый участок программы.

В организации цикла можно выделить следующие этапы:

- подготовка (инициализация) цикла (И);
- выполнение вычислений цикла (тело цикла) (Т);
- модификация параметров (М);
- проверка условия окончания цикла (У).

Порядок выполнения этих этапов, например, Т и М, может изменяться. В зависимости от расположения проверки условия окончания цикла различают циклы с нижним и верхним окончаниями. Для цикла с нижним окончанием тело цикла выполняется как минимум один раз, так как сначала производятся вычисления, а затем проверяется условие выхода из цикла. В случае цикла с верхним окончанием тело цикла может не выполниться ни разу в случае, если сразу соблюдается условие выхода.



Цикл называется детерминированным, если число повторений тела цикла заранее известно или определено. Цикл называется итерационным, если число повторений тела цикла заранее неизвестно, а зависит от значений параметров (некоторых переменных), участвующих в вычислениях.

## Раздел 5. Алгоритмизация и программирование

### Тема 5.2. Языки программирования

Язык программирования — это искусственный язык, отличающийся от естественных ограниченным числом "слов", значение которого понятно транслятору, и строгими правилами записи команд (операторов). Совокупность таких требований образует синтаксис языка программирования, а смысл каждой команды и других конструкций языка — его семантику.

С помощью языка программирования создается не готовая программа, а только её текст, описывающий ранее разработанный алгоритм. Чтобы получить работающую программу, надо этот текст либо автоматически перевести в машинный код (компиляция) и затем использовать отдельно от исходного текста, либо сразу выполнять команды языка, указанные в тексте программы (интерпретация).

Компиляторы обрабатывают весь текст программы (исходный код). Они просматривают его в поисках синтаксических ошибок, выполняют определённый смысловой анализ и затем транслируют (переводят) на машинный язык — генерируют машинный код. При этом часто выполняется оптимизация кода.

Интерпретатор анализирует и осуществляет пооператорную обработку и выполнение исходной программы или запроса без трансляции всей программы.

В реальных системах перемешаны технологии компиляции и интерпретации.

#### Уровни языков программирования

Различают языки высокого и низкого уровня.

Если язык программирования ориентирован на конкретный тип процессора и учитывает его особенности, то он называется языком низкого уровня. Имеется в виду, что операторы языка близки к машинному коду и ориентированы на конкретные процессоры.

Язык самого низкого уровня – ассемблер. Представляет каждую команду машинного кода с помощью условных символьных обозначений, называемых мнемониками. С помощью языков низкого уровня создаются компактные и эффективные программы, так как разработчик получает доступ ко всем командам процессора. Однако это требует совершенного знания компьютера, затрудняет отладку, а результирующая программа не может быть перенесена на компьютер с другим типом процессора. Подобные языки применяются для написания небольших системных приложений, драйверов.

Язык программирования высокого уровня ближе и понятнее человеку, нежели компьютеру. Особенности конкретных процессоров в них не учитываются, и поэтому результаты работы могут быть свободно перенесены на другие платформы. Разработка программы на языках высокого уровня значительно проще, а ошибок при создании значительно меньше.

### **Поколения языков программирования**

Языки программирования по своим возможностям и времени создания принято делить на несколько поколений (Generation Language, GL). К сегодняшнему дню насчитывается пять поколений языков. Каждое из последующих поколений языков по своей функциональной мощности качественно отличается от предыдущего.

В первое поколение входят языки, созданные в 40-50-х годах, когда компьютеры только появились на свет. В то время программы писались в машинных кодах, т. е. каждая компьютерная команда вместе с ее операндами вводилась в ЭВМ в двоичном виде. Это требовало огромных усилий по набору цифровых текстов и приводило к множеству трудноуловимых ошибок. В начале 50-х годов был написан первый ассемблер. Он позволял задавать названия команд в символьческом виде и указывать числа не только в двоичном, но и в десятичном или шестнадцатеричном формате, что существенно облегчало работу программистов. Языки первого поколения продолжают использоваться и сегодня, хотя в значительно меньшем объеме. Чаще всего приходится писать программы в машинных кодах для новых микропроцессоров, для которых еще не разработаны компиляторы, поддерживающие требуемый набор команд.

Расцвет второго поколения языков программирования (2 GL) пришёлся на конец 50-х – начало 60-х гг. Был создан символический ассемблер, позволявший писать программы без привязки к конкретным адресам памяти. В него было введено понятие переменной, и он, по сути, стал первым настоящим языком программирования со своим компилятором. Скорость разработки и эффективность функционирования программ резко возросли. Ассемблеры активно применяются в настоящее время, как правило, для создания программ, максимально использующих возможности аппаратуры – различных драйверов, модулей состыковки с нестандартным оборудованием и т. д. В некоторых областях, например в машинной графике, на ассемблере пишутся библиотеки, эффективно реализующие стандартные алгоритмы обработки изображений. Кроме того, среди программистов просто есть немало людей, предпочитающих использовать ассемблер в своей работе.

Появление третьего поколения принято относить к 60-м годам. В это время родились языки, которые называют универсальными языками высокого уровня, с их помощью можно решать задачи из любых областей. Тенденция к созданию универсальных языков была обусловлена реализацией идеи стандартизации ЭВМ и независимости программ от конкретной модели вычислительной машины. Это общеизвестные Fortran, Cobol, Algol, Pascal и др. Такие качества новых языков, как относительная простота, независимость от конкретного компьютера и возможность использования мощных синтаксических конструкций, позволили резко повысить производительность труда программистов. Кроме того, понятная большинству пользователей процедурная идеология этих языков привлекла к написанию небольших программ (как правило, расчетного или учетного характера) значительным количеством специалистов из некомпьютерных областей.

Подавляющее большинство языков 3 GL успешно применяется и сегодня. Современные компиляторы с интегрированными средами разработки предоставляют очень удобные средства поддержки процесса создания программ, легко осваиваемые студентами первых курсов, благодаря чему простые в освоении языки третьего поколения используются для разработки программ абсолютным большинством людей, зарабатывающих на жизнь программированием. Практически все современные коммерческие продукты, рассчитанные на массовый рынок, написаны на языках третьего поколения.

С начала 70-х годов по настоящее время продолжается период языков четвертого поколения. Происходит активное развитие систем программирования с ориентацией на реализацию крупных проектов, повышение надежности и скорости создания программных продуктов. Они обычно ориентированы на специализированные области применения, где хороших результатов добиваются, используя проблемно-ориентированные языки. Целью создания языков 4 GL было стремление увеличить скорость разработки проектов, уменьшить число ошибок и повысить общую надежность работы больших программных комплексов, получить возможность быстро и легко вносить изменения в готовые проекты, активно внедрять технологии визуальной разработки и т.д. Все языки четвертого поколения интегрированы в мощные пользовательские оболочки и обладают простым и удобным интерфейсом. Они чаще всего используются для проектирования баз данных и работы с ними (встроенные языки СУБД), что объясняется возможностью формализации всех понятий, используемых при построении баз данных. Языки 4 GL активно применяются в различных специализированных областях, где высоких результатов можно добиться, используя проблемно-ориентированные языки, оперирующие конкретными понятиями узкой предметной области. Как правило, в эти языки встраиваются мощные примитивы, позволяющие в одном операторе описать такую функциональность, для реализации которой на языках младших поколений потребовались бы тысячи строк кода.

Однако тем, кто использует языки 4 GL для создания законченных приложений, по-прежнему необходимо кодировать программу вручную с последовательным вводом команд. При этом сохраняется главный недостаток языков предыдущих поколений: ориентированы на чуждую человеческому мышлению чисто компьютерную идеологию (работа с памятью, переменными, базами данных, последовательностями абстрактных операторов и т. п.), что требует от людей хорошего понимания принципов функционирования компьютера и ОС.

Рождение языков пятого поколения относится к середине 90-х годов. Сюда относят не только новые мощные языки, но и системы создания программ, ориентированные на непрограммиста. Подобные системы отличаются стремлением предоставить конечному пользователю-неспециалисту богатые возможности создания прикладных программ с помощью визуальных средств разработки без знания программирования. Главная идея, которая закладывается в эти системы, – возможность компьютерного интерактивного или полностью автоматического преобразования инструкций, вводимых в компьютер, наиболее удобными человеку методами в максимально наглядном виде, в текст на универсальных языках программирования, описывающий готовую программу. Автоматизация процесса получения результирующего кода программы имеет следствиями невысокую эффективность (это связано с внутренней сложностью таких систем и стремлением разрабатывать независимые от платформы продукты), но повышенную надежность и отсутствие ошибок.

Системы пятого поколения имеют открытую архитектуру и поддерживают большое количество продуктов третьих фирм, предоставляя пользователю возможность интеграции с готовыми решениями для различных областей. Это могут быть всевозможные визуальные редакторы, генераторы отчетов, стандартные библиотеки, удобные мастера быстрого создания типовых приложений, CASE-системы, средства интеграции с базами данных и т. п. Чем больше приложений удастся объединить в одном пакете, тем большими возможностями он будет обладать.

Понятно, что системы этого поколения ориентируются на предоставление возможностей специалисту в области программирования эффективного средства разработки приложений для выполнения поставленных задач.

Существует, однако, и другое мнение на этот счет. Заключается оно в том, что языки пятого поколения являются именно языками программирования, требующими от разработчика соответствующей квалификации и умения составлять программы вручную. Сторонники этого мнения под языками 5GL понимают специализированные языки, оперирующие не абстрактными переменными, а понятиями своей предметной области. Это, как правило, узкоспециализированные языки, предоставляющие программисту мощные высокоуровневые возможности обработки информации из конкретной области знаний.

Программные комплексы с собственными языками программирования, интегрированные с базами знаний и экспертными системами, также относят к языкам пятого поколения.

К перспективным направлениям развития систем 5 GL следует отнести логическое программирование, объектно-ориентированное программирование, исследовательское программирование, технологии управления базами знаний, методы обработки и анализа текстовой информации (энциклопедии, Web-страницы, документы) с возможностью смыслового поиска и т.д. Для создания достаточно сложных приложений планируется использовать различные экспертные системы и базы знаний со встроенными языками логического программирования, позволяющие автоматизировать многие рутинные процессы и помочь пользователю найти правильный путь решения тех или иных задач.

### **Сравнительные характеристики языков программирования**

#### **Элементы объектной модели**

Каждый стиль программирования имеет свою концептуальную базу. Каждый стиль требует своего умонастроения и способа восприятия решаемой задачи. Для объектно-ориентированного стиля концептуальная база — это объектная модель. Она имеет четыре главных элемента: абстрагирование; инкапсуляция; модульность; иерархия.

Эти элементы являются главными в том смысле, что без любого из них модель не будет объектно-ориентированной. Кроме главных, имеются еще три дополнительных элемента: типизация; параллелизм; сохраняемость.

Называя их дополнительными, мы имеем в виду, что они полезны в объектной модели, но не обязательны.

Без такой концептуальной основы вы можете программировать на языке типа Smalltalk, Object Pascal, C++, CLOS, Eiffel или Ada, но из-под внешней красоты будет выглядывать стиль FORTRAN, Pascal или C. Выразительная способность объектно-ориентированного языка будет либо потеряна, либо искажена. Но еще более существенно, что при этом будет мало шансов справиться со сложностью решаемых задач.

#### **Абстрагирование**

Абстрагирование является одним из основных методов, используемых для решения сложных задач. Абстракция выделяет существенные характеристики некоторого объекта, отличающие его от всех других видов объектов и, таким образом, четко определяет его концептуальные границы с точки зрения наблюдателя.

#### **Инкапсуляция**

Абстракция и инкапсуляция дополняют друг друга: абстрагирование направлено на наблюдаемое поведение объекта, а инкапсуляция занимается внутренним устройством. Чаще всего инкапсуляция выполняется посредством скрытия информации, то есть маскировкой всех внутренних деталей, не влияющих на внешнее поведение. Обычно скрываются и внутренняя структура объекта и реализация его методов.

Инкапсуляция — это процесс отделения друг от друга элементов объекта, определяющих его устройство и поведение; инкапсуляция служит для того, чтобы изолировать контрактные обязательства абстракции от их реализации.

Инкапсуляция скрывает детали реализации объекта.

Модульность

Правильное разделение программы на модули является почти такой же сложной задачей, как выбор правильного набора абстракций.

Модули выполняют роль физических контейнеров, в которые помещаются определения классов и объектов при логическом проектировании системы. Такая же ситуация возникает у проектировщиков бортовых компьютеров. Логика электронного оборудования может быть построена на основе элементарных схем типа НЕ, И-НЕ, ИЛИ-НЕ, но можно объединить такие схемы в стандартные интегральные схемы (модули), например, серий 7400, 7402 или 7404.

Модульность позволяет хранить абстракции раздельно.

В традиционном структурном проектировании модульность — это искусство раскладывать подпрограммы по кучкам так, чтобы в одну кучку попадали подпрограммы, использующие друг друга или изменяемые вместе. В объектно-ориентированном программировании ситуация несколько иная: необходимо физически разделить классы и объекты, составляющие логическую структуру проекта.

Модульность — это свойство системы, которая была разложена на внутренне связанные, но слабо связанные между собой модули.

Иерархия

Значительное упрощение в понимании сложных задач достигается за счет образования из абстракций иерархической структуры. Определим иерархию следующим образом:

Иерархия — это упорядочение абстракций, расположение их по уровням.

Наследование — создание новых объектов из уже существующих. Начиная с определения самых общих абстрактных объектов, можно создавать более конкретные объекты нижнего уровня, которые не только унаследуют все функции своих предшественников, но могут добавлять им свои собственные. Принцип наследования позволяет упростить выражение абстракций, делает проект менее громоздким и более выразительным.

Типизация

Типизация — это способ защититься от использования объектов одного класса вместо другого, или по крайней мере управлять таким использованием.

Типизация заставляет нас выражать наши абстракции так, чтобы язык программирования, используемый в реализации, поддерживал соблюдение принятых проектных решений.

Идея согласования типов занимает в понятии типизации центральное место. Например, возьмем физические единицы измерения. Деля расстояние на время, мы ожидаем получить скорость, а не вес. В умножении температуры на силу смысла нет, а в умножении расстояния на силу — есть. Все это примеры сильной типизации, когда прикладная область накладывает правила и ограничения на использование и сочетание абстракций.

Сильная типизация заставляет нас соблюдать правила использования абстракций, поэтому она тем полезнее, чем больше проект. Однако у нее есть и теневая сторона. А именно, даже небольшие изменения в интерфейсе класса требуют перекомпиляции всех его подклассов. Кроме того, не имея параметризованных классов трудно представить себе, как можно было бы создать собрание разнородных объектов.

Полиморфизм означает, что разные объекты могут описывать различные реализации одного и того же метода.

Строгая типизация предотвращает смешивание абстракций.

Параллелизм

Есть задачи, в которых автоматические системы должны обрабатывать много событий одновременно. В других случаях потребность в вычислительной мощности превышает ресурсы одного процессора. В каждой из таких ситуаций естественно использовать несколько компьютеров для решения задачи или задействовать многозадачность на многопроцессорном компьютере. Процесс (поток управления) — это фундаментальная единица действия в системе. Каждая программа имеет по крайней мере один поток управления, параллельная система имеет много таких потоков: век одних недолог, а другие живут в течении всего сеанса работы системы. Реальная параллельность достигается только на многопроцессорных системах, а системы с одним процессором имитируют параллельность за счет алгоритмов разделения времени.

Параллелизм — это свойство, отличающее активные объекты от пассивных.

Параллелизм позволяет различным объектам действовать одновременно.

Сохраняемость любой программный объект существует в памяти и живет во времени. Аткинсон предположил, что есть непрерывное множество продолжительности существования объектов: существуют объекты, которые присутствуют лишь во время вычисления выражения, но есть и такие, как базы данных, которые существуют независимо от программы. Этот спектр сохраняемости объектов охватывает:

## **Раздел 5. Алгоритмизация и программирование**

### **Тема 5.3. Стили программирования**

Одним из важнейших признаков классификации языков программирования является принадлежность их к одному из стилей, основными из которых являются следующие стили:

- процедурный,
- функциональный,
- логический

- объектно-ориентированный.

Стиль программирования - набор приемов или методов программирования, которые используют программисты, чтобы получить правильные, эффективные, удобные для применения и легко читаемые программы.

Стиль программирования, под которым понимают стиль оформления программ и их «структурность», также существенно влияет на читаемость программного кода и количество ошибок программирования.

Основные принципы форматирования

Основная теорема форматирования гласит, что хорошее визуальное форматирование показывает логическую структуру программы. Создание красивого кода — хорошо, а демонстрация структуры кода — лучше. Если одна методика лучше показывает структуру кода, а другая выглядит красивой, следует использовать ту, которая лучше демонстрирует структуру.

Форматирование — это ключ к структуре программы. Компьютеру важна исключительно информация о скобках или операторах `begin` и `end`, а читатель-человек склонен делать выводы из визуального представления кода.

Способы форматирования

Можно получить хороший формат кода, поразному используя несколько инструментов для форматирования.

Неотображаемые символы. Используйте неотображаемые символы для улучшения читаемости. Неотображаемые символы, к которым относятся пробелы, знаки табуляции, переводы строк и пустые строки, — это основное средство для демонстрации структуры программы.

Группировка взаимосвязанных выражений — еще один способ применения неотображаемых символов. В литературе мысли группируются в абзацы. Хорошо написанный абзац содержит предложение, относящиеся только к определенной идее. Он не должен содержать посторонних предложений. Точно так же абзац кода должен содержать только взаимосвязанные операторы, выполняющие одно задание.

Пустые строки. Кроме необходимости группировать взаимосвязанные операторы, очень важно отделять несвязанные выражения друг от друга. Начало нового абзаца в книге обозначается отступом или пустой строкой. Начало нового абзаца в коде нужно указывать с помощью пустой строки.

Пустые строки позволяют продемонстрировать организацию программы. Вы можете использовать их для деления групп взаимосвязанных операторов на абзацы, отделения методов друг от друга и выделения комментариев.

Хотя эту статистику тяжело применить на практике, но одно исследование показало, что оптимальное число пустых строк в программе составляет от 8 до 16 %. Если оно больше 16 %, то время, затрачиваемое на отладку, заметно увеличивается.

Отступы. Применяйте отступы для демонстрации логической структуры программы. Как правило, операторы выделяются отступами, когда они следуют после некоторого выражения, от которого они логически зависят. Оптимальными являются отступы из 2—4 пробелов.

Скобки. Используйте скобки чаще, чем вам это кажется необходимым. Применяйте скобки для разъяснения выражений, состоящих из двух и более членов.

### **Процедурное программирование**

Процедурное (императивное) программирование является отражением архитектуры традиционных ЭВМ, которая была предложена фон Нейманом в 40-х годах. Теоретической моделью процедурного программирования служит алгоритмическая система под названием «машина Тьюринга».

Программа на процедурном языке программирования состоит из последовательности операторов (инструкций), задающих процедуру решения задачи. Основным является оператор присваивания, служащий для изменения содержимого областей памяти. Концепция памяти как хранилища значений, содержимое которого может обновляться операторами программы, является фундаментальной в императивном программировании.

Выполнение программы сводится к последовательному выполнению операторов с целью преобразования исходного состояния памяти, то есть значений исходных данных, в заключительное, то есть в результаты. Таким образом, с точки зрения программиста имеются программа и память, причем первая последовательно обновляет содержимое последней.

Процедурные языки характеризуются следующими особенностями:

- необходимостью явного управления памятью, в частности, описанием переменных;
- малой пригодностью для символьных вычислений;
- отсутствием строгой математической основы;
- высокой эффективностью реализации на традиционных ЭВМ.

Одним из важнейших классификационных признаков процедурного языка является его уровень. Уровень языка программирования определяется семантической (смысловой) емкостью его конструкций и степенью его ориентации на программиста. Язык программирования частично ликвидирует разрыв между методами решения различного рода задач человеком и вычислительной машиной. Чем более язык ориентирован на человека, тем выше его уровень. Дадим краткую характеристику реализованным на ПЭВМ языкам программирования в порядке возрастания их уровня.

Двоичный язык является непосредственно машинным языком. В настоящее время такие языки программистами практически не применяются.

Язык Ассемблера — это язык, предназначенный для представления в удобочитаемой символической форме программ, записанных на машинном языке. Он позволяет программисту пользоваться мнемоническими

кодами операций, присваивать удобные имена ячейкам и областям памяти, а также задавать наиболее удобные схемы адресации.

Язык Макроассемблера является расширением языка Ассемблера путем включения в него макросредств. С их помощью в программе можно описывать последовательности инструкций с параметрами — макроопределения. После этого программист может использовать снабженные аргументами макрокоманды, которые в процессе ассемблирования программы автоматически замещаются макрорасширениями. Макрорасширение представляет собой макроопределение с подставленными вместо параметров аргументами.

Другими словами, язык Макроассемблера представляет средства определения и использования новых, более мощных команд как последовательности базовых инструкций, что несколько повышает его уровень.

Языки Ассемблера и Макроассемблера применяются системными программистами-профессионалами с целью использования всех возможностей оборудования ЭВМ и получения эффективной по времени выполнения и по требуемому объему памяти программы. На этих языках обычно разрабатываются относительно небольшие программы, входящие в состав системного программного обеспечения: драйверы, утилиты и другие.

Язык программирования С (Си) первоначально был разработан для реализации операционной системы UNIX в начале 70-х годов. В последующем приобрел высокую популярность среди системных и прикладных программистов. В настоящее время этот язык реализован на большинстве ЭВМ.

В С сочетаются достоинства современных высокоуровневых языков в части управляющих конструкций и структур данных с возможностями доступа к аппаратным средствам ЭВМ на уровне, который обычно ассоциируется с языком низкого уровня типа языка Ассемблера. Язык С имеет синтаксис, обеспечивающий краткость программы, а компиляторы способны генерировать эффективный объектный код.

Одна из наиболее существенных особенностей С состоит в нивелировании различий между выражениями и операторами, что приближает его к функциональному языку. В частности, выражение может обладать побочным эффектом присваивания, а также может использоваться в качестве оператора. Нет также четкой границы между процедурами и функциями, более того, понятие процедуры не вводится вообще.

Синтаксис языка затрудняет программирование и восприятие составленных программ. Отсутствует и строгая типизация данных, что предоставляет дополнительные возможности программисту, но не способствует написанию надежных программ.

Basic (BeginnersAll-purposeSymbolicInstructionCode) — многоцелевой язык символических инструкций для начинающих) представляет собой простой язык программирования, разработанный в 1964 году для использования новичками. Он был разработан как простейший язык для непосредственного общения человека с вычислительной машиной. Поэтому первоначально работа велась в интерактивном режиме с использованием интерпретаторов. В настоящее время для этого языка имеются также и компиляторы.

Согласно концепциям, заложенным в Basic, этот язык в смысле строгости и стройности является антиподом языка Pascal. В частности, в нем широко распространены различные правила умолчания, что считается плохим тоном в большинстве языков программирования подобного типа.

Basic широко распространен на ЭВМ различных типов и очень популярен в среде программистов, особенно начинающих. Существует множество диалектов этого языка, мало совместимых между собой. Basic активно поглощает многие концепции и новинки из других языков. Поэтому он достаточно динамичен, и нельзя однозначно определить его уровень.

Pascal является одним из наиболее популярных среди прикладных программистов процедурным языком программирования, особенно для ПЭВМ. Разработанный в 1970 году швейцарским специалистом в области вычислительной техники профессором Н. Виртом, язык назван в честь французского математика и по замыслу автора предназначался для обучения программированию. Однако язык получился настолько удачным, что стал одним из основных инструментов прикладных и системных программистов при решении задач вычислительного и информационно-логического характера. В 1979 году был подготовлен проект описания языка — Британский стандарт языка программирования PascalBS6192, который стал также и международным стандартом ISO 7185.

### **Функциональное программирование**

Сущность функционального (аппликативного) программирования определена А. П. Ершовым как «... способ составления программ, в которых единственным действием является вызов функции, единственным способом расчленения программы на части является введение имени для функции, а единственным правилом композиции — оператор суперпозиции функции. Никаких ячеек памяти, ни операторов присваивания, ни циклов, ни, тем более, блок-схем, ни передачи управления».

Роль основной конструкции в функциональных языках играет выражение. К выражениям относятся скалярные константы, структурированные объекты, функции, тела функций и вызовы функций. Функция трактуется как однозначное отображение из  $X^N$  в  $X$ , где  $X$  — множество выражений.

Аппликативный язык программирования включает следующие элементы:

- классы констант, которыми могут манипулировать функции;
- набор базовых функций, которые программист может использовать без предварительного объявления и описания;
- правила построения новых функций из базовых;
- правила формирования выражений на основе вызовов функций.

Программа представляет собой совокупность описаний функций и выражения, которые необходимо вычислить. Данное выражение вычисляется посредством редукции, то есть серии упрощений, до тех пор, пока это

возможно по следующим правилам: вызовы базовых функций заменяются соответствующими значениями; вызовы небазовых функций заменяются их телами, в которых параметры замещены аргументами.

Функциональное программирование не использует концепцию памяти как хранилища значений переменных. Операторы присваивания отсутствуют, вследствие чего переменные обозначают не области памяти, а объекты программы, что полностью соответствует понятию переменной в математике. В принципе, можно составлять программы и вообще без переменных. Кроме того, нет существенных различий между константами и функциями, то есть между программами и данными. В результате этого функция может быть значением вызова другой функции и может быть элементом структурированного объекта. Число аргументов при вызове функции не обязательно должно совпадать с числом параметров, указанных при ее описании. Перечисленные свойства характеризуют аппликативные языки как языки программирования очень высокого уровня.

Первым таким языком был LISP (LISTProcessing — обработка списков), созданный в 1959 году. Цель его создания состояла в организации удобства обработки символьной информации. Существенная черта этого языка — унификация программных структур и структур данных: все выражения записываются в виде списков.

#### **Логическое программирование**

Новую область — логическое, или реляционное программирование, — открыло появление языка PROLOG {Пролог} (PROgramminginLOGic — программирование в терминах логики). Этот язык был создан французским ученым А. Кольмером в 1973 году. В настоящее время известны и другие языки, однако наиболее развитым и распространенным языком логического программирования является именно Пролог. Так, имеется свыше 15 различных его реализации на ПЭВМ. Языки логического программирования, в особенности Пролог, широко используются в системах искусственного интеллекта, рассматриваемых в данном учебном пособии.

Центральным понятием в логическом программировании является отношение. Программа представляет собой совокупность определенных отношений между объектами (в терминах условий или ограничений) и цели (запроса). Процесс выполнения программы трактуется как процесс общезначимости логической формулы, построенной из программы по правилам, установленным семантикой используемого языка. Результат вычисления является побочным продуктом этого процесса. В реляционном программировании нужно только специфицировать факты, на которых алгоритм основывается, а не определять последовательность шагов, которые требуется выполнить. Это свидетельствует о декларативности языка логического программирования. Она метко выражена в формуле Р. Ковальского: «алгоритм = логика + управление». Языки логического программирования характеризуются:

- высоким уровнем;
- строгой ориентацией на символьные вычисления;
- возможностью инверсных вычислений, то есть переменные в процедурах не делятся на входные и выходные;
- возможной логической неполнотой, поскольку зачастую невозможно выразить в программе определенные логические соотношения, а также невозможно получить из программы все выводы правильные.

Логические программы, в принципе, имеют небольшое быстродействие, так как вычисления осуществляются методом проб и ошибок, поиском с возвратами к предыдущим шагам.

## **Раздел 5. Алгоритмизация и программирование**

### **Тема 5.4. Понятие об объектно-ориентированном программировании**

В основе объектно-ориентированного языка программирования лежат два основных понятия: объект и класс. Основными характеристическими свойствами этих понятий являются:

Инкапсуляция - комбинирование записей с процедурами и функциями, манипулирующими полями этих записей, формирует новый тип данных - объект (под записью понимается переменная типа "запись").

Наследование - определение объекта и его дальнейшее использование для построения иерархии порожденных объектов с возможностью для каждого порожденного объекта, относящегося к иерархии, доступа к коду и данным всех порождающих объектов.

Полиморфизм - присваивание действию одного имени, которое затем совместно используется вниз и вверх по иерархии объектов, причем каждый объект иерархии выполняет это действие способом, именно ему подходящим.

Объект - как базовое понятие в объектно-ориентированном программировании

Понятию "объект" сопоставляют ряд дополняющих друг друга определений. Ниже приведены некоторые из них.

Объект - это осязаемая реальность, характеризующаяся четко определяемым поведением.

Объект - особый опознаваемый предмет, блок или сущность (реальная или абстрактная), имеющая важное функциональное назначение в данной предметной области.

Объект может быть охарактеризован структурой, его состоянием, поведением и индивидуальностью. Состояние объекта определяется пересечением всех возможных (обычно статических) свойств и текущими значениями (обычно динамическими) каждого из этих свойств. Свойства объекта характеризуются значениями его параметров. Поведение объекта описывает, как объект воздействует на другие объекты или как он подвергается воздействию со стороны других объектов с точки зрения изменения его собственного состояния и состояния других объектов. Говорят также, что поведение объекта определяется его действиями. Определенное воздействие одного объекта на другой с целью вызвать соответствующую реакцию называют операцией. В объектно-ориентированных языках программирования операции называют методами.

Можно выделить пять типов операций:

- конструктор, создание и инициализация объекта;
- деструктор, разрушающий объект;
- модификатор, изменяющий состояние объекта;
- селектор для доступа к переменным объекта без их изменения;
- итератор для доступа к содержанию объекта по частям в определенной последовательности.

Известна и другая классификация методов объекта, когда выделяют функции управления, реализации, доступа и вспомогательные функции.

Под индивидуальностью объекта понимают свойство объекта, позволяющее отличать этот объект от всех других объектов.

Объекты могут находиться в определенных отношениях друг к другу. Эти отношения могут быть иерархическими. Основные иерархические отношения - это отношения использования и включения. Отношение использования реализуется посылкой сообщений от объекта А к объекту В. При этом объект А может выступать в роли:

- активного или воздействующего объекта, когда он воздействует на другие объекты, но сам воздействию не подвергается;
- пассивного или исполняющего, когда объект подвергается воздействию, но сам на другие объекты не воздействует;
- посредника, если объект и воздействует и сам подвергается воздействию.

Отношение включения имеет место, когда составной объект содержит другие объекты.

### **Инкапсуляция**

Объекты моделируют характеристики и поведение элементов мира, в котором мы живем. Они являются окончательной абстракцией данных.

Объекты содержат вместе все свои характеристики и особенности поведения. Отношения частей к целому и взаимоотношения между частями становятся понятнее тогда, когда все содержится вместе в одной упаковке. Это и называется инкапсуляцией.

Наследование как важнейшее свойство объекта

Не менее важным является и тот факт, что объекты могут наследовать характеристики и поведение того, что мы называем порождающие, родительские объекты (или предки). Здесь происходит качественный скачок: наследование, возможно, является сегодня единственным самым крупным различием между обычным программированием на Паскале и объектно-ориентированным программированием в Borland Pascal.

В терминах Паскаля, объект во многом схож с записью, которая является оболочкой для объединения нескольких связанных элементов под одним именем.

Процесс, с помощью которого один тип наследует характеристики другого типа, называется наследованием. Наследник называется порожденным (дочерним) типом, а тип, которому наследует дочерний тип, называется порождающим (родительским) типом. Ранее известные типы записей Паскаля не могут наследовать. Однако Borland Pascal расширяет язык Паскаля для поддержки наследования. Одним из этих расширений является новая категория структуры данных, связанная с записями, но значительно более мощная. Типы данных в этой новой категории определяются с помощью нового зарезервированного слова `object`. Тип объекта может быть определен как полный, самостоятельный тип в манере описания записей Паскаля, но он может определяться и как потомок существующего типа объекта путем помещения порождающего (родительского) типа в скобки после зарезервированного слова `object`.

Большая часть конструирования объектно-ориентированных прикладных программ состоит в построении иерархии объектов, являющейся отражением генеалогического дерева объектов в приложениях.

Как показали данные определения, объекты тесно связаны с записями. Новое зарезервированное слово `object` является наиболее очевидным различием, но как мы увидим позднее, имеется большое число других различий, некоторые из которых довольно тонкие.

Экземпляры объектных типов

Экземпляры объектных типов описываются в точности так же, как в Паскале описывается любая переменная, либо статическая, либо указатель, ссылающийся на размещенную в динамической памяти переменную.

### **Поля объектов**

Мы можем обратиться к полю объекта в точности так же, как к полю обычной записи, либо с помощью оператора `with`, либо путем уточнения имени с помощью точки.

### **Методы**

Даже если мы можем обратиться к полям объекта непосредственно, это будет не совсем хорошей идеей. Принципы объектно-ориентированного программирования требуют, чтобы поля объектов были исключены из исходного кода, насколько это возможно. Это ограничение поначалу может показаться спорным и жестким, но оно является только частью огромной картины объектно-ориентированного программирования.

Ответ заключается в том, что при всякой возможности для доступа к полям данных должны использоваться методы объекта. Метод является процедурой или функцией, описанной внутри объекта и жестко ограниченной этим объектом.

Методы являются одними из наиболее примечательных атрибутов объектно-ориентированного программирования и требуют некоторой практики перед использованием.

Метод - это процедура или функция, объединенная с данным типом столь тесно, что метод является как бы окруженным невидимым оператором `with`, что делает экземпляр данного типа доступными изнутри для метода. Определение типа включает заголовок метода. Полное определение метода квалифицируется в имени ти-

па. Тип объекта и метод объекта являются двумя лицами этой новой разновидности структуры, именуемой методом.

Одним из важнейших принципов объектно-ориентированного программирования является то, что программист во время разработки программы должен думать о коде и о данных совместно. Ни код, ни данные не существуют в вакууме. Данные управляют потоком кода, а код манипулирует образами и значениями данных.

Если наш код и данные являются разделенными элементами, то всегда существует опасность вызова правильной процедуры с неверными данными или ошибочной процедуры с правильными данными. Забота о совпадении этих элементов возлагается на программиста, и хотя строгая типизация Паскаля здесь помогает, самое лучшее, что он может сделать - это указать на несоответствие.

Объект осуществляет синхронизацию кода и данных путем совместного построения их описаний. Реально, чтобы получить значение одного из полей объекта, мы вызываем относящийся к этому объекту метод, который возвращает значение нужного поля. Чтобы присвоить полю значение, мы вызываем метод, который назначает данному полю новое значение.

Однако, Borland Pascal не вынуждает нас делать это. Как всякое структурное программирование, объектно-ориентированное программирование является дисциплиной, которую мы должны навязать себе, используя предоставляемые языком средства. Borland Pascal позволяет нам обращаться к полям объекта непосредственно извне объекта, однако он поощряет нас использовать преимущества объектно-ориентированного программирования и создавать методы для манипулирования полями объекта внутри самого объекта.

Объект состоит из структуры данных и связанных с ней процедур (которые называются методами), которые работают с данными, записанными в экземплярах структуры данных.

Объект может наследовать характеристики порождающего объекта. Это означает, что структура данных нового объекта включает структуру данных порождающего объекта, а также новые данные. Кроме того, новый объект может вызывать все процедуры порождающего объекта, а также те процедуры методов, которые в нем описываются.

Объект, не имеющий наследования, называется базовым объектом. Объект, наследующий характеристики других объектов, называется порожденным или производным объектом.

#### Полиморфизм

Полиморфизм означает, что один и тот же метод выполняется по-разному для различных объектов. Например, метод класса Музыкальный инструмент - PlayMusicForAnOrchestra (играй музыку для оркестра) - может быть определен как общий метод, который может использоваться с любой категорией музыкальных инструментов. Этот метод написан таким образом, что не важно, какой именно инструмент получает задание играть, однако для классов, описывающих конкретные инструменты, данный метод должен быть переопределен (override), что даст возможность определить конкретные действия, учитывающие особенности данного инструмента.

#### Понятие класса

В класс объединяются объекты с одинаковыми свойствами и методами.

Одним из первых действий, предпринимаемых человеком при попытке понять окружающий мир, является применение к нему некоторой структурной формы. При встрече с неизвестным объектом мы пытаемся втиснуть его в нашу существующую структуру: другими словами, классифицировать его. Большинство людей знакомо по крайней мере с несколькими классификационными структурами или иерархиями.

Использование иерархии классов вводит необходимость абстракции. Классы становятся более абстрактными по мере продвижения вверх по иерархии.

Объектно-ориентированные языки используют такой же подход. Иерархии обычно начинаются с нескольких абстрактных классов. Каждый новый класс представляется как подкласс существующего класса (называемого его суперклассом). Он наследует данные и методы от классов, стоящих выше в иерархии. Только те данные и методы, которые являются новыми для этого класса, следует определить и реализовать.

Класс - это абстрактное понятие, сравнимое с понятием категория в его обычном смысле.

По определенным свойствам любого элемента определенной категории можно установить, что он принадлежит к этой категории. Сама категория определяется общими свойствами, которые имеют все экземпляры этой категории.

Это можно пояснить на примере музыкальных инструментов. Музыкальные инструменты делятся на следующие категории: духовые, ударные и струнные.

Все эти категории принадлежат к категории музыкальных инструментов. В свою очередь, категория музыкальных инструментов входит в категорию инструментов. На рис.1 эта структура категорий графически представлена в виде дерева.

Музыкальные инструменты имеют общие свойства, но каждый инструмент сам по себе обладает особыми свойствами, которые определяют его назначение и отличают его от других инструментов. По тому же принципу можно описать и классы в ООП. Определенный музыкальный инструмент некоторой категории, например труба, является объектом. Категория, к которой этот инструмент принадлежит, - это класс.

Класс в объектно-ориентированном программировании - это абстрактный тип данных, который включает не только данные, но и функции и процедуры.

Функции и процедуры класса называются методами и содержат исходный код, предназначенный для обработки внутренних данных объекта данного класса.

Наличие в классах абстрагирования, которое максимально удобно для программистов, работающих с этими классами, имеет первостепенное значение при разработке повторно используемого ПО. Если вы выстро-

ите интерфейс, на который не влияют изменения в реализации, то вашему приложению долгое время не понадобятся никакие модификации. Если вы знакомы с предметной областью задачи, вы без труда определите, какие методы понадобятся пользователям класса. Скажем так: если при проектировании класса вам удастся сочетать хорошее знание предметной области с прогнозом относительно дальнейших перспектив использования класса, можно гарантировать, что большая часть интерфейса этого класса останется неизменной, даже в случае возможного совершенствования реализации класса. В результате отстранения пользователя от деталей реализации система в целом становится понятнее, а значит, и удобнее в работе.

Между классами также могут быть установлены отношения:

- отношение разновидности (кошка - вид определенного биологического семейства или кошка - домашнее животное);
- включения или составной части (лапа - часть кошки);
- ассоциативности, когда между классами есть чисто смысловая связь (кошки и собаки - домашние животные).

#### **Процесс объектно-ориентированного проектирования**

Объектно-ориентированное проектирование (Object-Oriented Design - OOD) - это поступательный итеративный процесс. Граница между объектно-ориентированным анализом и проектированием расплывчата и построение проекта программного изделия состоит из ряда циклов, в которых уточняются описания классов и взаимодействия между ними, разрабатываются реализующие их программы, проводится их отладка и тестирование и по результатам каждого этапа уточняются рабочие документы предыдущих этапов, дорабатываются описания классов и программы. Эти циклы повторяются до получения требуемого результата.

В рассмотренном выше примере были выделены классы “множество данных” и “данное”. Пусть классу “множество данных” присвоено имя TXSet.

С учетом имеющихся инструментальных средств класс TXSet может быть построен на основе класса Array из библиотеки CLASSLIB, т.е. это множество может быть интерпретировано массивом. Массив представляет собой упорядоченную совокупность однотипных элементов, в то же время данные могут принадлежать различным типам и каждому типу соответствует свой набор характеристик. Это противоречие можно преодолеть, если элементами массива TXSet будут указатели на экземпляры данных.

Чтобы использовать указатели на экземпляры данных как элементы массива, все классы, определяющие типы данных, должны быть образованы из общего базового класса.

Пусть требуется обеспечить возможность использования числовых скалярных данных и массивов (векторов и прямоугольных матриц), а также данных типа строк и массива строк. Естественно определить для каждого такого типа свой класс: TDScal, TDArray, TDString, TDStringArray. В каждом из этих классов должно быть поле идентификатора данного ident, поле описания данного head и, возможно, поле flags, представляющее собой набор битов, дополняющих описание данного. Может оказаться удобным иметь и поля, содержащие количество знаков при представлении скаляра или элементов массивов (width) и количество цифр в дробной части для представления чисел (dec). Все эти данные можно объединить в классе TData, базовом для остальных классов данных. Таким образом, вместо одного класса “данное”, выделенного на этапе анализа, появилось пять классов. После этого следует вернуться к этапу анализа и оформить рабочие документы анализа для новых классов.

Аналогичным образом следует уточнить состав и определения остальных классов, выбранных на этапе анализа.

После определение перечня классов следует разработать семантику каждого класса - определить состав и назначение методов класса. При этом также может возникнуть необходимость выделения новых классов и, следовательно, повторение отдельных частей этапа анализа и новое уточнение ранее описанных классов.

Таким образом, процесс объектно-ориентированного проектирования состоит из циклического выполнения четырех основных шагов:

- Определение классов и объектов на определенном уровне абстракции.
- Определение семантики классов.
- Определение (идентификация) связей между классами и объектами.
- Реализация классов.

На каждом повторении этого цикла уточняются описания классов и перерабатываются проектные документы.

### **4.3. Лабораторные работы**

<i>№ n/n</i>	<i>Номер раздела дисциплины</i>	<i>Наименование лабораторной работы</i>	<i>Объем (час.)</i>	<i>Вид занятия в ин- терактивной, активной, инновационной формах, (час.)</i>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>

1	4.	Основы работы в ОС Windows.	1	Разбор конкретных ситуаций (1 час.)
2	4.	Microsoft Word. Основы работы [1,2].	0,5	-
3	4.	Microsoft Word. Форматирование абзацев [1,2].	0,5	-
4	4.	Microsoft Word. Работа с таблицами [1,2].	0,5	-
5	4.	Microsoft Word. Редактор формул [1,2].	-	-
6	4.	Microsoft Excel. Основы работы [1,2].	0,5	-
7	4.	Microsoft Excel. Использование ссылок [1,2].	0,5	-
8	4.	Microsoft Excel. Построение диаграмм [1,2].	0,5	-
9	4.	Microsoft Power Point. Создание презентаций.	1	Проектная деятельность (1 час.)
10	5.	QBASIC. Ввод и вывод данных. Программирование алгоритмов линейной структуры [6].	1	Проектная деятельность (1 час.)
11	5.	QBASIC. Программирование алгоритмов разветвленной структуры [6].	-	-
12	5.	QBASIC. Программирование циклических алгоритмов [6].	1	Проектная деятельность (1 час.)
13	5.	QBASIC. Работа с массивами [6].	0,5	-
14	5.	QBASIC. Работа с символьными переменными [6].	0,5	-
15	5.	QBASIC. Работа с файлами данных [6].	-	-
<b>ИТОГО</b>			<b>8</b>	<b>4</b>

#### 4.4. Практические занятия

Учебным планом не предусмотрены.

#### 4.5. Контрольные мероприятия: контрольная работа.

**Цель:** Целью работы является приобретение навыков практического применения приложений пакета Microsoft Office.

**Структура:** 8 разделов: 1 – Рабочий стол Windows; 2 – Скоростной набор текста; 3 – Форматирование различных частей документа; 4 – Работа в редакторе формул Word; 5 – Построение графиков в Excel; 6 – Форматирование графиков в Excel ; 7 – Проведение автоматизированных расчётов в Excel; 8 – Проведение расчётов и построение графиков в программе MathCAD.

**Основная тематика:** В контрольной работе предлагается сформировать фрагмент текста с применением различных стилей форматирования, таблиц и формул; выполнить расчет с применением электронных таблиц.

**Рекомендуемый объем:** Контрольная работа выполняется в виде отчета объёмом 5...15 страниц.

Выдача задания и проверка контрольной работы проводится в соответствии с календарным учебным графиком

**5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

<i>Компетенции</i> <i>№, наименование</i> <i>разделов дисциплины</i>	<i>Кол-во</i> <i>часов</i>	<i>Компетенции</i>	$\Sigma$ <i>комп.</i>	<i>t</i> <sub>ср</sub> , <i>час</i>	<i>Вид</i> <i>учебной</i> <i>работы</i>	<i>Оценка</i> <i>результатов</i>
		<i>ОПК-1</i>				
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>1.</b> Общая характеристика процессов сбора, передачи, обработки и накопления информации.	16	+	1	16	Лк, СРС	Экзамен
<b>2.</b> Локальные и глобальные сети ЭВМ.	18	+	1	18	Лекции, СРС	Экзамен
<b>3.</b> Технические средства реализации информационных процессов.	18	+	1	18	Лекции, СРС	Экзамен
<b>4.</b> Программные средства реализации информационных процессов.	53	+	1	53	Лекции, ЛР, СРС	кн, экзамен
<b>5.</b> Алгоритмизация и программирование.	30	+	1	30	Лекции, ЛР, СРС	кн, экзамен
<b><i>всего часов</i></b>	<b>135</b>	<b>135</b>	<b>1</b>	<b>135</b>		

## 6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Лабораторный практикум по информатике: Учеб. пособие для вузов / В.С. Микшина, Г.А. Еремеева, К.И. Бушмелева и др.; Под ред. В.А. Острейковского. – 2-е изд., стереотип. – М.: Высш. школа, 2006. – 376 с.
2. Михеева Е.В. Практикум по информатике: Учеб. пособие / Е.В. Михеева. – 5-е изд., стереотип. – М.: Академия, 2007. – 192 с.
3. Васильева С.А. Информатика: Методические указания к курсовой работе / С.А.Васильева.– Братск: БрГУ, 2009. – 22 с.

## 7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№	Наименование издания	Вид занятия	Количество экземпляров в библиотеке, шт.	Обеспеченность, (экз./ чел.)
1	2	3	4	5
<b>Основная литература</b>				
1.	Информатика. Базовый курс: учебник / Под ред. С.В. Симоновича. – 2-е изд. – СПб.: Питер, 2014. – 640 с	Лк,СР, ЛР	76	1
2.	Информатика: учебник / А.Б. Золотов, П.А. Акимов [и др.]. – М.: АСВ, 2010. – 336 с.	Лк,СР, ЛР	25	1
<b>Дополнительная литература</b>				
3.	Самохина М.И. Работа с СУБД Microsoft Access: Учебное пособие / М.И. Самохина, Н.А. Барковская. – Братск: БрГУ, 2008. – 85 с.	СР	49	1
4.	Кузин А.В. Базы данных: Учеб. пособие для вузов /А.В. Кузин, С.В. Левонисова. – 3-е изд., стереотип. – М.: Академия, 2008. – 320 с.	КР, СР	30	1
5.	Кудрявцев Е.М. Начальное знакомство с компьютерными системами Word, Mathcad, КОМПАС: учебное пособие / Е.М. Кудрявцев. – М.: АСВ, 2007. – 160 с.	СР	25	1
6.	Баженова И.Ю. Введение в программирование: Учеб. пособие для вузов / И.Ю. Баженова, В.А. Сухомлин. – М.: ИНТУИТ.РУ, 2007. – 326 с.	ЛР, СР	5	0,3

## 8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО - ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронный каталог библиотеки БрГУ  
[http://irbis.brstu.ru/CGI/irbis64r\\_15/cgiirbis\\_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=](http://irbis.brstu.ru/CGI/irbis64r_15/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=).
2. Электронная библиотека БрГУ  
<http://ecat.brstu.ru/catalog>.
3. Электронно-библиотечная система «Университетская библиотека online» <http://biblioclub.ru>.
4. Электронно-библиотечная система «Издательство «Лань»  
<http://e.lanbook.com>.
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru>.
6. Научная электронная библиотека eLIBRARY.RU <http://elibrary.ru>.
7. Университетская информационная система РОССИЯ (УИС РОССИЯ) <https://uisrussia.msu.ru/>.
8. Национальная электронная библиотека НЭБ  
<http://xn--90ax2c.xn--p1ai/how-to-search/>.

## 9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Практическое владение компьютером предполагает наличие сформированных знаний по конструкции частей ПЭВМ, а также умений и навыков работы с различными чаще используемыми программами.

Для того чтобы достигнуть указанного в целевой установке уровня владения компьютером, следует систематически готовиться к занятиям, выполнять в полном объеме все задания лабораторных работ и закреплять полученные умения, повторяя пройденный на занятиях материал во время самостоятельной подготовки.

### 9.1. Методические указания для обучающихся по выполнению лабораторных работ

#### Лабораторная работа №1. Основы работы в ОС Windows.

Цель работы – освоить навыки работы с манипулятором «мышь», ознакомиться с основными элементами интерфейса ОС Microsoft Windows, научиться пользоваться встроенным средством получения справочной информации.

Для краткости очередность выбора пунктов и/или кнопок будем обозначать в виде последовательности имен этих кнопок, разделенных символом \ (обратный слеш).

Например:

Пуск\Справка и поддержка,

Пуск\ Программы\ Стандартные.

При выборе каждого пункта необходим один или два щелчка левой кнопкой мыши. Для вывода подсказки на экран необходимо щелкнуть кнопку, помеченную надписью «Справка» или нажать клавишу F1. Для выполнения многих из всех возможных в ОС MS Windows операций пользуйтесь контекстными меню, которые для большинства объектов можно вызвать однократным нажатием правой кнопки мыши на этом объекте или области активного окна.

Порядок выполнения:

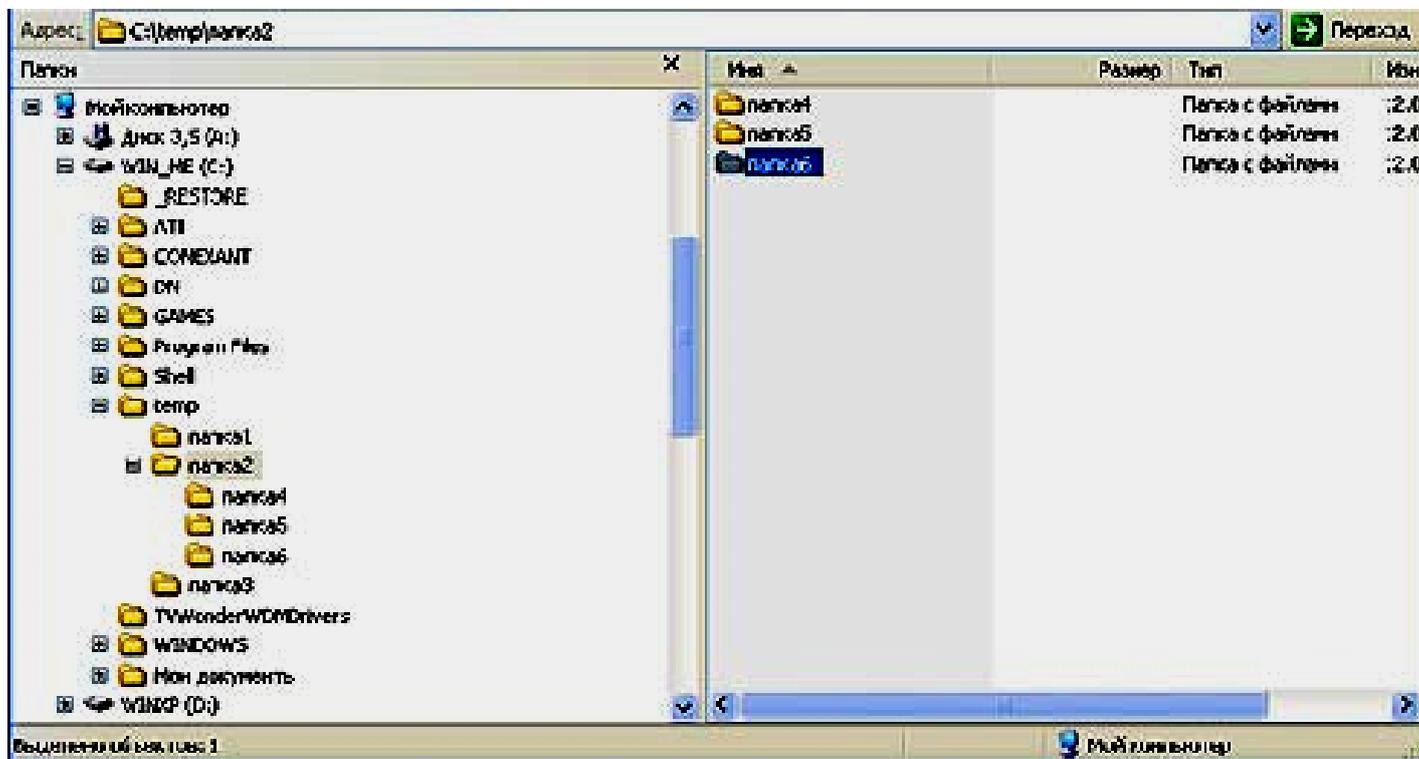
1. После запуска MS Windows щелкните левой кнопкой мыши кнопку с надписью «Пуск».
2. Щелкните по пункту «Справка и поддержка», а затем в появившемся окне справочной системы MS Windows выберите пункт «Основы работы в Windows», затем пункты, «Основные задачи, выполняемые системой Windows», «Советы по работе со справочной системой». В каждом пункте ознакомьтесь с перечнем предлагаемых для более подробного изучения тем.
3. Выберите пункт «Работа с файлами и папками». Ознакомьтесь с содержанием раздела и с представленной в нем информацией об основных операциях с папками и файлами.
4. Ознакомьтесь с содержанием пунктов «Поиск сведений» и «Защита компьютера».
5. Дважды щелкнув по ярлыку «Мой компьютер» на рабочем столе, ознакомьтесь с его назначением и возможностями, для чего в окне поиска «Центра справки и поддержки» укажите пункт «Мой компью-

тер», ознакомьтесь с относящимися к нему подпунктами. Запустите Проводник, щелкнув ярлык «Проводник», находящийся в подпунктах меню кнопки «Пуск»: Пуск\Все программы\Стандартные\Проводник. Сравните интерфейс Проводника и папки «Мой компьютер».

6. Изучите возможности Проводника при поиске, переименовании и перемещении файлов. Изучите контекстные меню в панелях Проводника.

7. Ознакомьтесь с пунктами основного меню Проводника (строка в верхней части окна).

8. Создайте на диске С в папке “Temp” (путь C:\Temp) папку, в которой создайте другие, как показано на рисунке:

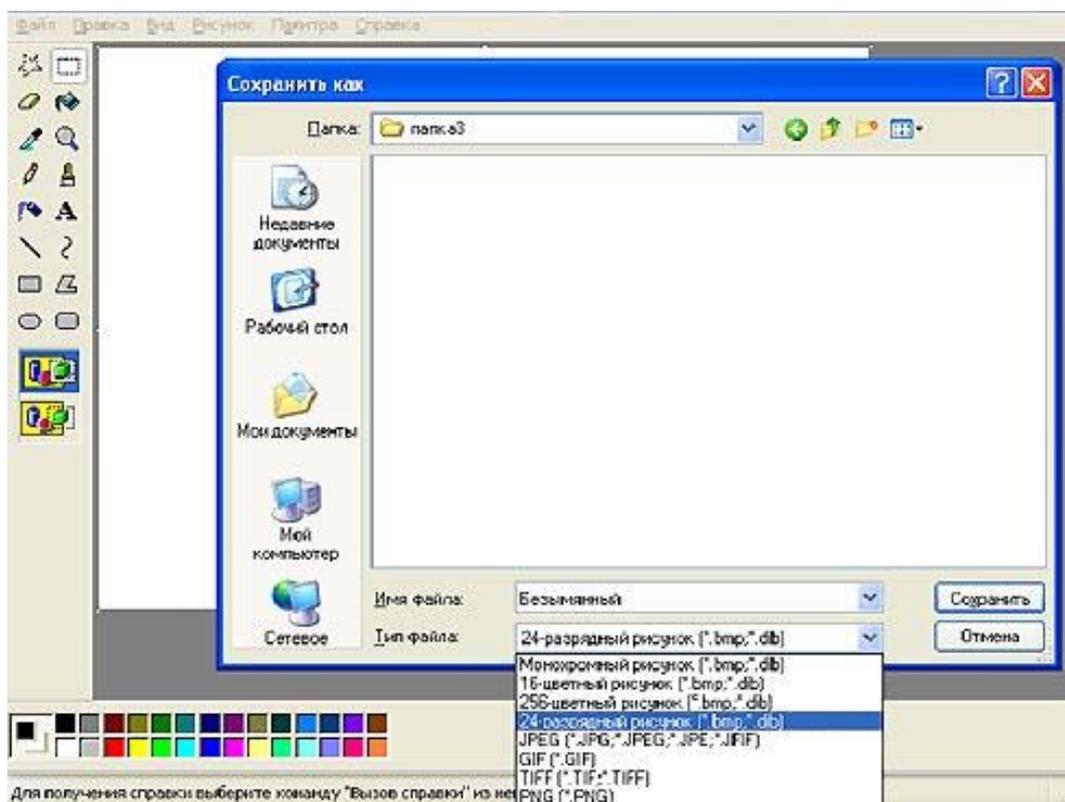


9. Создайте текстовый файл в Папке 1 с помощью текстового редактора WordPad, для этого запустите его (Пуск\Программы\Стандартные\WordPad). Ознакомьтесь с пунктами меню «Файл», «Правка», «Вид», «Вставка», «Формат», используя при затруднениях справку. Сохраните этот файл под разными именами в Папке 3, и под одинаковыми в Папках 5 и 6. Для этого ознакомьтесь с соответствующими разделами справочной службы MS Windows.

10. Скопируйте в Папку 4 все файлы из какой-либо другой папки (не менее 20 файлов). Используя пункт меню «Вид», расположите на экране файлы разными способами. Скопируйте Папку 1 в Папку 5. Переместите Папку 1 из Папки 5 в Папку 3. Используйте для этого возможности Проводника и папки Мой компьютер.

11. Создайте файл с изображением новогодней елочки с помощью графического редактора Paintbrush, для этого запустите его, выбрав ярлык в пунктах кнопки «Пуск»: Пуск\Программы\Стандартные\Paint. Сохраните его в Папке 3 под одним именем в нескольких разных графических форматах (выбрать в основном меню редактора Файл\Сохранить как..., далее в появившемся диалоговом окне сохранения указать имя и формат сохраняемого файла). В окне Проводника тремя разными способами узнайте размер сохраненных файлов с изображениями.

12. Покажите преподавателю результат выполнения работы на экране монитора.



Форма ОТЧЕТНОСТИ: отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

Основная литература [1,2].

Контрольные вопросы для самопроверки

1. Что такое папка?
2. Что такое панель задач?
3. Поясните назначение кнопки пуск?
4. Что такое Рабочий стол?
5. Поясните назначение папки Мой компьютер?
6. Поясните назначение программы Проводник?
7. Как получить информацию о свойствах объекта?
8. Назовите кнопки управления размерами окна?
9. Какие программы относятся к числу стандартных программ Windows?
10. Назовите основные приемы копирования файлов и папок?
11. Назовите основные приемы выделения объектов?
12. Для чего служит главное меню Windows, как производится его настройка?
13. Как происходит запуск программ в Windows?
14. Как с помощью панели задач, управлять окнами программ и открытыми папками на рабочем столе?
15. Как используется контекстное меню?

**Лабораторная работа №2. Microsoft Word. Основы работы.**

Цель работы – освоить навыки работы с клавиатурой, ознакомиться с основными элементами интерфейса Microsoft Word научиться пользоваться основными средствами оформления документов.

Порядок выполнения:

**1. Создать новый документ и набрать абзац текста:**

Наиболее распространенными неисправностями системы питания дизельных двигателей являются износ и регулировка плунжерных пар насоса высокого давления и форсунок, потеря герметичности этих агрегатов. Возможны также износ выходных отверстий форсунки, их закоксование и засорение. Эти неисправности приводят к изменению момента начала подачи топлива, неравномерности работы топливного насоса по углу и количеству подаваемого топлива, ухудшению качества

распыливания топлива форсункой. В результате перечисленных неисправностей повышается расход топлива и увеличивается токсичность отработавших газов. Диагностическими признаками неисправностей системы питания являются: затруднение пуска двигателя; расхода топлива под нагрузкой; падение мощности двигателя и его перегрев; изменение состава и повышение токсичности отработавших газов. Диагностика систем питания дизельных двигателей проводится методами ходовых и стендовых испытаний и оценки состояния механизмов и узлов системы после их демонтажа. Диагностирование системы питания дизельных двигателей включает в себя проверку герметичности системы и состояния топливных и воздушных фильтров, проверку топливо подкачивающего насоса, а также насоса высокого давления и форсунок. Герметичность системы питания, дизельного двигателя имеет особое значение. Так, подсос воздуха во впускной части системы приводит к нарушению работы топливоподающей аппаратуры.

**2. Предварительно сохранить документ под именем Лаб\_1\_Фамилия (рекомендуется производить сохранение через каждые 10 мин. работы).**

**3. Провести проверку орфографии и при необходимости отредактировать текст.**

**4. Скопировать набранный абзац 2 раза (получится 3 абзаца).**

**5. Для первого абзаца задать заголовок "Исходный текст".**

**6. Для второго абзаца задать заголовок "Отступы" и установить следующие параметры:**

a) Отступы: слева - 2,5см, справа - 4,5см.

b) Выравнивание – по ширине

c) Первая строка – нет

d) Шрифт – Arial, 13 пт, курсив, синий

**7. Для третьего абзаца задать заголовок "Красная строка" и установить следующие параметра форматирования:**

a) Выравнивание – по ширине

b) Первая строка – отступ (1,25см)

c) Межстрочный интервал – двойной

d) Интервал до абзаца (отбивка) – 12 пт

e) Шрифт – Courier New, 10 пт, обычный, черный

**8. Все заголовки к абзацам отцентрировать и установить для них следующие параметры форматирования:**

a) Шрифт- Times New Roman, 20 пт, полужирный

b) Задать регистр – все прописные буквы

**9. Заключить текст второго абзаца в рамку и заполнить фоном:**

a) Рамка: двойная линия, ширина 1,5 пт, черная, с тенью

b) Заливка: узор 30%, цвет зеленый

**10. Установить следующие параметры страницы:**

a) Поля: верхнее - 2см, нижнее – 2,5см, левое – 3см, правое – 1,5см

b) Ориентация – книжная

c) Размер бумаги – стандарт А4

**11. Создать колонтитулы:**

a) В верхнем колонтитуле указать Фамилию, группу (разместить все по центру)

b) В нижнем колонтитуле указать дату (по правому полю)

**12. Установить нумерацию страниц: внизу страницы, по центру, с указанием номера на первой странице.**

**13. Окончательно сохранить документ.**

**14. Осуществить предварительный просмотр документа.**

**Напечатать документ**

**Форма отчетности:** отчет по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчет по лабораторному практикуму.

Основная литература [1,2].

Контрольные вопросы для самопроверки

1. Что такое командная строка?

2. Что такое строка прокурукты документа?

3. Как сохранить документ?

**Лабораторная работа №3. Microsoft Word. Форматирование абзацев.**

**Цель работы** – подготовить документ, содержащий набор абзацев с одинаковым форматированием (в абзацах есть красная строка и установлено выравнивание по ширине). В тексте есть ряд новых для вас объектов: сноски, списки и рисунок, сделанный в программе Microsoft Draw.

**Порядок выполнения:**

## 1. Набор текста

Перед тем, как приступит к вводу текста, в пустом документе установите поля размером по 2 см с каждой стороны (с помощью команды **Файл**⇒**Параметры страницы**). Теперь введите текст по образцу:

### Характеристика предприятия АТП-2 и его деятельности

Enter

Автотранспортное предприятие №2 (АТП-2) входит в состав ОАО «Братскэнергостройтранс-1» (ОАО «БЭСТ-1»).

Enter

Предприятие в основном занимается коммерческими перевозками леса, осуществляет перевозку навалочных грузов, а также осуществляет пассажирские перевозки работников ОАО «БЭСТ-1» на работу и с работы.

Enter

Предприятие расположено на промышленной площадке Падунского района. Застройка предприятия павильонная, то есть производственные корпуса находятся отдельно друг от друга. Здания содержатся в удовлетворительном состоянии.

Enter

АТП-2 ОАО «БЭСТ-1» является смешанным предприятием, то есть в парке имеются грузовые автомобили (бортовые, специальные, специализированные, самосвалы), легковые автомобили и автобусы. АТП-2 ОАО «БЭСТ-1» специализируется на перевозке грузов, поэтому основную часть парка составляют грузовые автомобили.

По численности автомобилей АТЦ относительно предприятиям средней мощности с  $A_{н}$  более 100 ед. и представляет собой АТП комплексного типа. На нем организованы следующие виды технических воздействий: ТО-1, ТО-2, ТР, Д-2, СО. Также предприятие осуществляет хранение подвижного состава.

Enter

Режим работы – 250 дней в году в одну смену, продолжительностью 8 часов. Режим работы подразделений следующий: подвижной состав работает в одну смену продолжительностью 8 часов 354 дня в году; зона ТО – одна смена по 8 часов 250 дней в году; работники ИТС, диспетчерская служба, автослесари зоны ТР работают в одну смену по 8 часов 250 дней в году.

Enter

Чтобы не тратить много времени на набор текста, а сосредоточиться на форматировании документа, добавим большую часть оставшегося текста из файла-заготовки. Выберите команду **Вставка**⇒**Файл**. Найдите в папке «Мои документы» файл «Фрагмент текста для лабораторной работы № 3.doc». Текст из этого файла будет добавлен в ваш документ (для обучающихся по дистанционной системе см. файл рядом в каталоге с данными методическими указаниями; можно воспользоваться режимом копировать/вставить).

Сохраните документ в файл с именем, содержащим название вашей группы и фамилии.

## 2. Шрифтовое оформление

Выделите весь текст (команда **Правка**⇒**Выделить все**). Присвойте выделенному фрагменту тексту шрифт **Times New Roman**, размер **13 пунктов**. На панели инструментов **Форматирование** в списке размеров нет значения 13. Чтобы установить именно этот размер, сделайте следующее:

1) Щелкните мышью внутри белого поля списка размеров (туда, где в результате должно быть число 13). Там появится курсор.

2) Наберите 13 с клавиатуры и нажмите [Enter].

3) Снимите выделение текста.

В соответствие с образцом, выделите полужирным шрифтом заголовки и название рисунка.

## 3. Оформление абзацев

Строки заголовков и названия рисунка отцентрируйте с помощью кнопки на панели инструментов **Форматирование**.

У остальных абзацев основного текста необходимо установить выравнивание по ширине и назначить красную строку. Выделите все абзацы поочередно, используя клавишу [Ctrl]. Кнопкой на панели инструментов установите выравнивание по ширине. Чтобы назначить красную строку, вызовите окно свойств абзацев командой **Формат**⇒**Абзац** в списке **Первая строка** выберите вариант **Отступ**, рядом установите значение 1,5 см.

Теперь измените межстрочный интервал. Выделите весь текст, выберите команду **Формат**⇒**Абзац** и в списке **Междустрочный** укажите **Полуторный**.

## 4. Расстановка переносов

В документах с большим количеством однородных абзацев для экономии пространства листа принято включать перенос слов. Этот параметр действует на весь документ, и для его установки выделять весь текст не нужно.

Чтобы включить перенос, выберите команду **Сервис**⇒**Язык**⇒**Расстановка переносов**, поставьте флажок **Автоматическая расстановка переносов**. Закройте окно кнопкой **ОК**.

## 5. Списки

Найдите в тексте фрагмент, который будет содержать список:

Основное производство включает выполнение работ по ежедневному обслуживанию (ЕО), ТО, ТР. Вспомогательное производство выполняет механические, тепловые, электротехнические, шиномонтажные и другие работы. Обслуживающее производство - обеспечивает работоспособность оборудования, зданий и сооружений АТП, а также включает в себя склады и мастерские ОГМ.

Структура технической службы включает в себя:

- производственно-технический отдел (ПТО);
- отдел главного механика (ОГМ);
- отдел технического контроля (ОТК);
- отдел материально-технического снабжения (ОМТС).

Плановый отдел составляет планы хозяйственной деятельности АТП на следующий период.

Бухгалтерия предприятия осуществляет учет, контроль и ревизию хозяйственной деятельности предприятия, на основе документов ведет расчет рентабельности и прибыли.

Инженер по БД контролирует техническую исправность ПС и отвечает за безопасность движения, находится непосредственно в АТП.

ПТО и ОМТС являются общими для всего ПО БЭСТ и не относятся непосредственно к АТП-2.

Требуется отформатировать следующим образом:

Структура технической службы включает в себя:

- 1) производственно-технический отдел (ПТО);
- 2) отдел главного механика (ОГМ);
- 3) отдел технического контроля (ОТК);
- 4) отдел материально-технического снабжения (ОМТС).

В тексте вместо списка присутствуют дефисы. Для создания автоматического списка выделите в тексте четыре строки перечисления. Выберите команду **Формат**⇒**Список**. В появившемся окне выберите **Нумерованный** в виде [ 1 ) ]. Отформатируйте положение списка также как текст перемещением указателей на верхней линейке. Удалите дефисы.

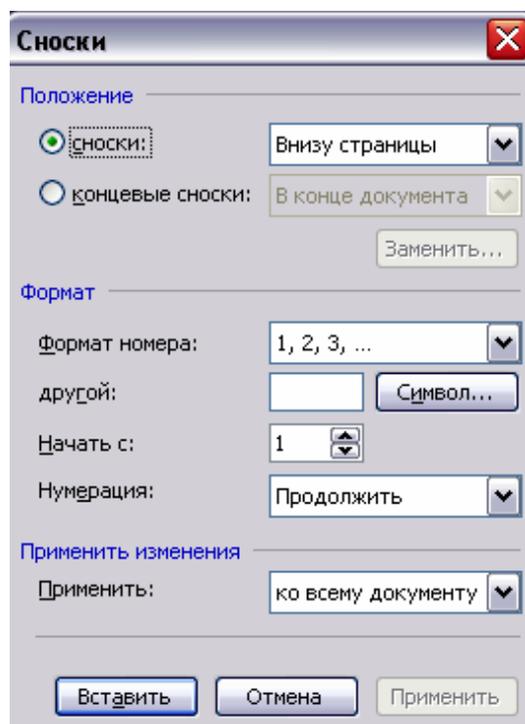
## 6. Сноски

В этом же фрагменте текста имеются четыре сноски. В образце основного текста номера сносок выглядят как надстрочные индексы. Текст сносок набран внизу листа под разделительной горизонтальной чертой. Это *обычные* сноски. Сноски бывают концевыми, текст которых набирают в конце документа.

Первую сноску сделайте следующим образом:

1) Поставьте курсор на второй странице текста после фрагмента: "1) производственно-технический отдел (ПТО)", после скобки, то есть там, где должен быть надстрочный индекс с номером сноски.

2) Выберите команду **Вставка**⇒**Ссылка**⇒**Сноска**. Появится диалоговое окно (Рисунок 1). Оставьте все параметры по умолчанию, нажмите кнопку **Вставить**.



3) Появится разделительная черта внизу листа, в месте набора текста сноски появится курсор. Наберите следующее:

ПТО разрабатывает мероприятия по повышению эффективности производства, снижению простоев автомобилей и расхода топлива, затрат труда и материалов и т.д.

Чтобы выйти из сноски, щелкните мышью в свободной части документа (**Enter нажимать не надо!**)

Аналогичным образом добавьте еще три сноски:

Сноска 2: после **(ОГМ)** с текстом:

ОГМ обеспечивает поддержание оборудования, зданий, сооружений и энергосилового хозяйства в технически исправном состоянии, а также монтаж и обслуживание производственного оборудования.

Сноска 3: после **(ОТК)** с текстом:

ОТК осуществляет контроль качества проведения работ, контролирует техническое состояние ПС.

Сноска 2: после **(ОМТС)** с текстом:

ОМТС обеспечивает постоянное снабжение АТП запчастями в нужном количестве и ассортименте, горюче-смазочными материалами (ГСМ), необходимым оборудованием и т.д.

## 7. Колонтитулы

Колонтитулы – это текст или рисунок, который помещается внизу или вверху страницы документа. В зависимости от места расположения колонтитул бывает верхний или нижний. Например, номера страниц.

Допускается создавать уникальный колонтитул для первой страницы документа или убирать колонтитул с первой страницы. Можно разделять колонтитул для четных и нечетных страниц.

В вашем документе верхним колонтитулом должна быть строка:

АТП-2 ОАО «Братскэнергостройтранс-1»

Эта строка должна быть набрана шрифтом **Times New Roman**, размер **11 пунктов**, выровнена по правому краю. Вставьте колонтитул следующим образом:

1) Выберите команду **Вид** ⇒ **Колонтитулы**. Вы перейдете в режим редактирования колонтитула, по умолчанию в верхний.

2) Наберите и отформатируйте текст колонтитула.

3) Выйти из колонтитула, можно дважды щелкнув на основном тексте.

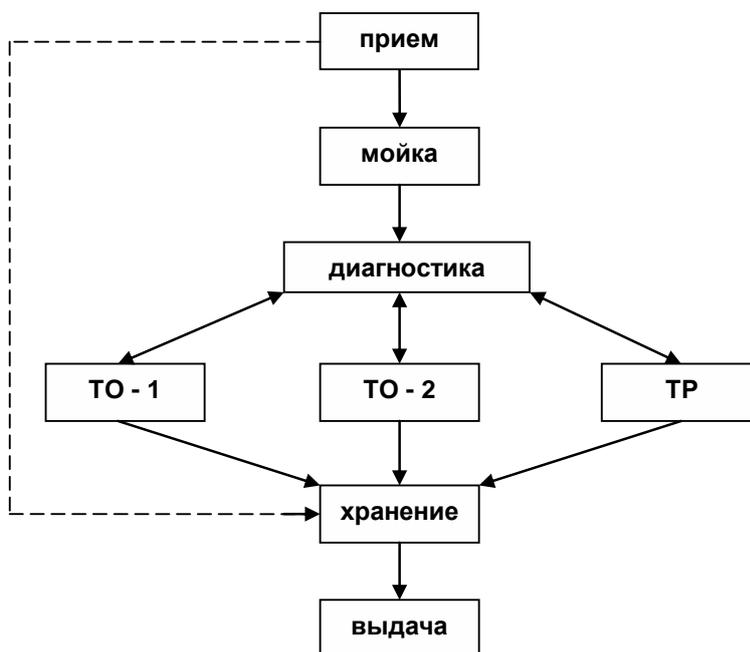
Нижним колонтитулом будут номера страниц. Можно создать их автоматически. Выберите команду **Вставка⇒Номера страниц**. Укажите расположение номеров **Внизу страницы, От центра**, флажок **Номер на первой странице** не выключать.

## 8. Рисунок Microsoft Draw

В документе есть рисунок. Он должен располагаться перед его названием:

Рисунок 1 – Схема организации технологического процесса ТО и ТР

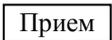
Поставьте курсор в строку, где должен быть рисунок. Выберите команду **Вставка⇒Объект⇒Рисунок Microsoft Word**. Будет запущена программа **Microsoft Draw**. Пока она работает документ не видно. Рисунок выполняется при помощи инструментов на панели **Рисование**. Для возврата в документ, когда рисунок готов, необходимо выбрать команду **Файл⇒Закреть и вернуться**.



Первую надпись в рамке создайте следующим образом:

- 1) На панели **Рисование** инструмент **Надпись** (кнопка ).
- 2) Нарисуйте прямоугольник мышью. Внутри него появится курсор.
- 3) Установите шрифт **Arial, 10 пунктов, полужирный**.
- 4) Наберите текст надписи – слово *прием*.
- 5) Отцентрируйте строку.
- 6) Мышью за угловой маркер подберите размер рамки так, чтобы слово было видно целиком.
- 7) Установите поля текста внутри надписи при помощи команды **Формат⇒надпись**, вкладка **Надпись**, значения полей выбрать так, чтобы размер надписи соответствовал рисунку.

Аналогичным образом сделайте еще 6 надписей и расположите их как на рисунке.

Соедините надписи   стрелкой:

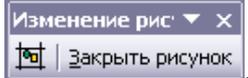
На панели **Рисование** инструмент **Стрелка** (кнопка ). Нарисуйте стрелку. Нарисуйте подобным образом все стрелки. Где необходимо поставить двойную стрелку (между Диагностикой и ТО-2), после обрисовки обычной стрелки выбрать на панели

**Рисование** вид стрелки, нажав на кнопку  **Вид стрелки**.

Штриховые линии рисуются при помощи вызова команды *Автофигуры* ⇒ *Линии* на панели **Ри-**

**сование**. Тип линии задается нажатием кнопки **Меню Штрих** .

Когда рисунок готов надо настроить его размер и затем вернуться в документ:

1) Найдите панель инструментов **Изменение рисунка** .

Если панель неактивна, вызовите ее командой

**Вид** ⇒ **Панель инструментов** ⇒ **Изменение рисунка**.

2) На панели нажмите кнопку **Восстановить границы рисунка** .

3) Нажмите кнопку **Заккрыть рисунок**.

4) В документе отцентрируйте рисунок.

Форма отчетности: отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

Основная литература [1,2].

Контрольные вопросы для самопроверки

1. Что такое отформатировать абзац?
2. Что такое строка форматирования?
3. Как установить расстояние между строк?

#### Лабораторная работа №4. Microsoft Word. Работа с таблицами.

Цель работы – получить навыки преобразования текста в таблицу и обратно, работы с таблицами, оформления таблиц, сортировки данных в таблице.

Порядок выполнения:

1. Создайте таблицу.

Затраты		
Отделы	Канцелярские товары	Расходные материалы на оргтехнику
Отдел снабжения	46	200
Отдел маркетинга	200	230
Плановый	120	400
Бухгалтерия	340	560
Отдел кадров	20	140
Технический отдел	373	749

**Примечание:** для созданной таблицы применить ручное форматирование.

2. Выделите вторую строку и выровняйте по центру. Также выровняйте по центру столбцы с числовыми данными.

3. Преобразовать рассматриваемую таблицу следующим образом:

– Вставить столбцы *Сумма* и *НДС* и оформить их как вычисляемые.

– Вставить строку *ИТОГО* и подсчитать в ней суммы по каждому столбцу.

Оформление столбца Сумма.

Значение этого столбца вычисляется как сумма столбцов *Канцелярские товары* и *Расходные материалы на оргтехнику* с помощью команды **Таблица** ⇒ **Формула**.

В появившемся окне в строке **Формула** нужно набрать =SUM(LEFT), что означает суммирование числовых ячеек, расположенных слева Формулу нужно набирать для каждой ячейки столбца.

Оформление столбца НДС.

Значение НДС определяется как 5% от значений столбца *Сумма*.

В этом случае столбцы таблицы имеют имена А,В,С,Д..., строки нумеруются 1,2,3,..., ячейки имеют адреса из имени столбца и номера строки, на пересечении которых они расположены – А1, В3, D4 и т.д., как в табличном редакторе.

Для подсчета значения в первой строке столбца *НДС* в окне команды **Таблица** ⇒ **Формула** набираем формулу = D3\*5%, для второй строки =D4\*5% и т.д..

Оформление строки ИТОГО.

Для вычисления суммы по столбцу *Канцелярские товары* нужно в окне команды **Таблица**⇒**Формула** набрать формулу =SUM(ABOVE) Аналогично вычисляются суммы по остальным столбцам.

*Расположение заголовка по центру.*

1. Выделить первую строку и выполнить команду **Таблица**⇒**Объединить ячейки**
2. Выровнять текст по центру с помощью соответствующей кнопки на панели форматирования. Созданная таблица должна иметь следующий вид:

Затраты (руб.)				
Отделы	Канцелярские товары	Расходные материалы на оргтехнику	Сумма	НДС
Отдел снабжения	46	200	246	12,3
Отдел маркетинга	200	230	430	21,5
Плановый	120	400	520	26
Бухгалтерия	340	560	900	45
Отдел кадров	20	140	160	8
Технический отдел	373	749	1122	56,1
<b>ИТОГО</b>	<b>1099</b>	<b>2279</b>	<b>3378</b>	<b>168,9</b>

Форма отчетности: отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

Основная литература [1,2].

Контрольные вопросы для самопроверки

1. Что такое отформатировать таблицу?
2. Что такое рисование таблицы?
3. Как скопировать таблицу и вставить её с целью редактирования?

### Лабораторная работа №5. Microsoft Word. Редактор формул.

Цель работы – получить навыки работы в редакторе формул Microsoft Equation 3. Научиться создавать сложные математические формулы в общепринятом понятном виде.

Порядок выполнения:

Редактор формул Microsoft Equation 3.0 предназначен для вставки в текстовые документы объектов в виде формул с использованием математических и специальных символов, например:

$$a_{02} = \frac{R_{X2}}{P_{Z2}} r_{e02}$$

Откройте приложение **Microsoft Word**, введите в начале документа свою фамилию, инициалы, группу, номер и название лабораторной работы.

Введите следующий текст:

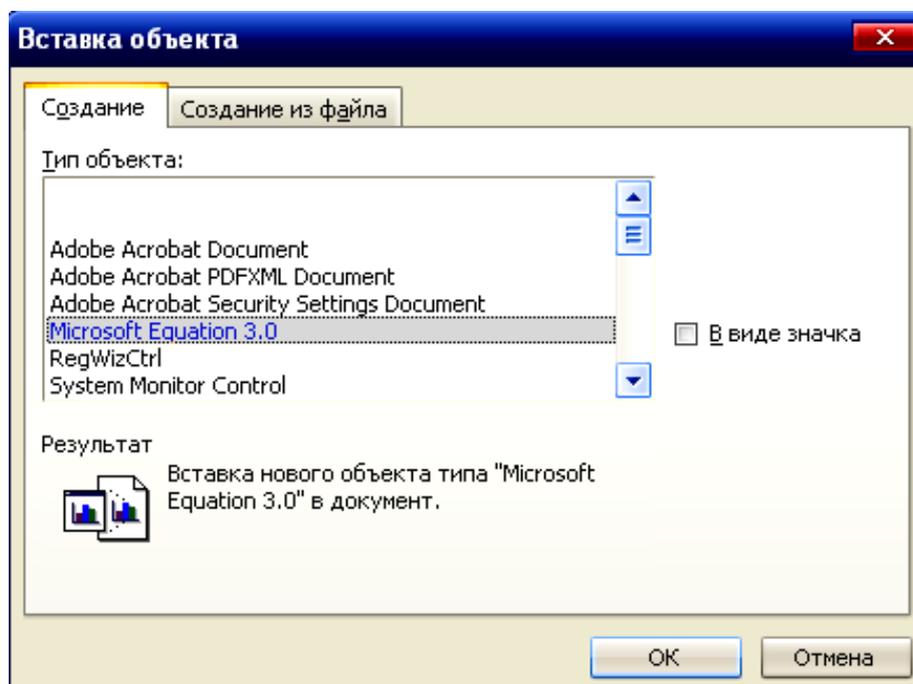
«Путевой расход топлива автомобиля:»

Затем необходимо вставить формулу, для этого:

1. Укажите место для вставки формулы (нажмите *Enter*).

2. В меню *Вставка* выберите команду *Объект*, а затем откройте вкладку

*Создание.*



3. В списке Тип объекта выберите Microsoft Equation 3.0.

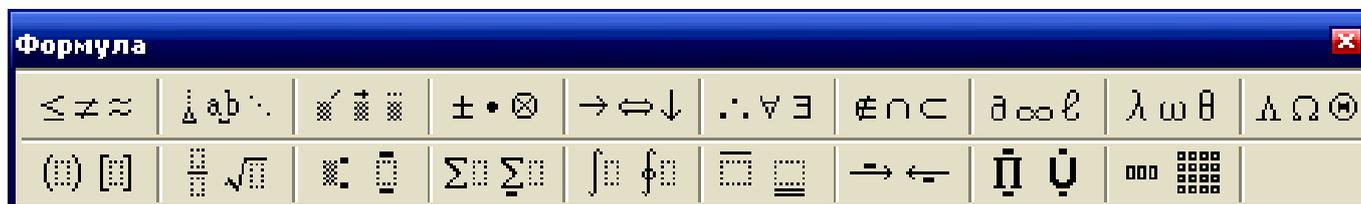
4. Нажмите кнопку ОК.

Если редактор формул недоступен (Элемент «Microsoft Equation 3.0» отсутствует в списке объектов в диалоговом окне «Объект»), его необходимо установить.

5. Создайте формулу

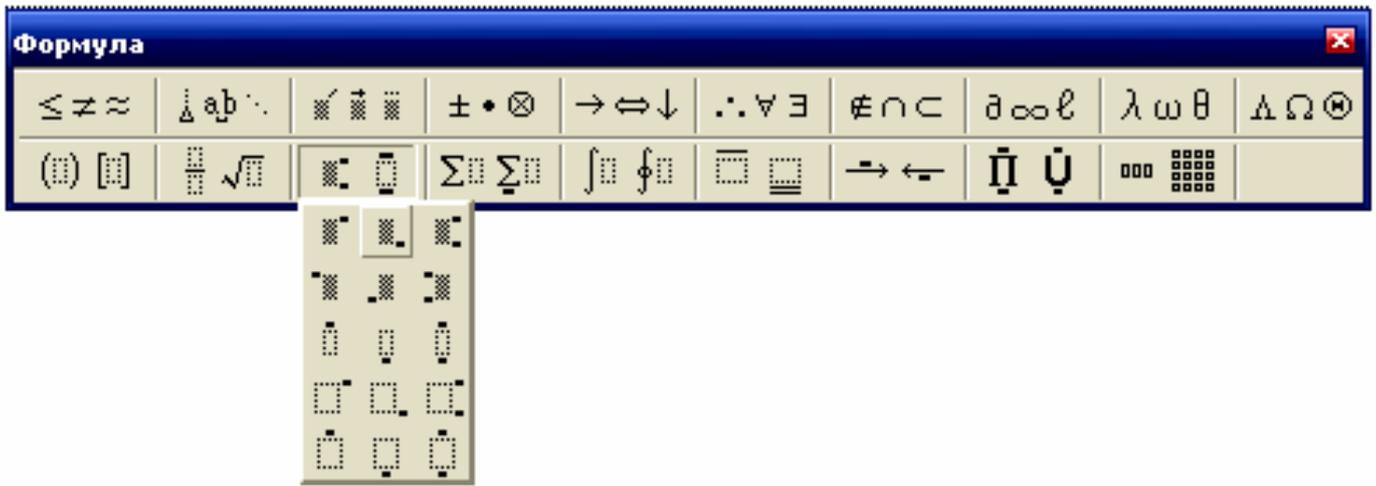
$$Q_S = 100 \frac{dQ}{dS},$$

выбирая символы на панели инструментов **Формула**,

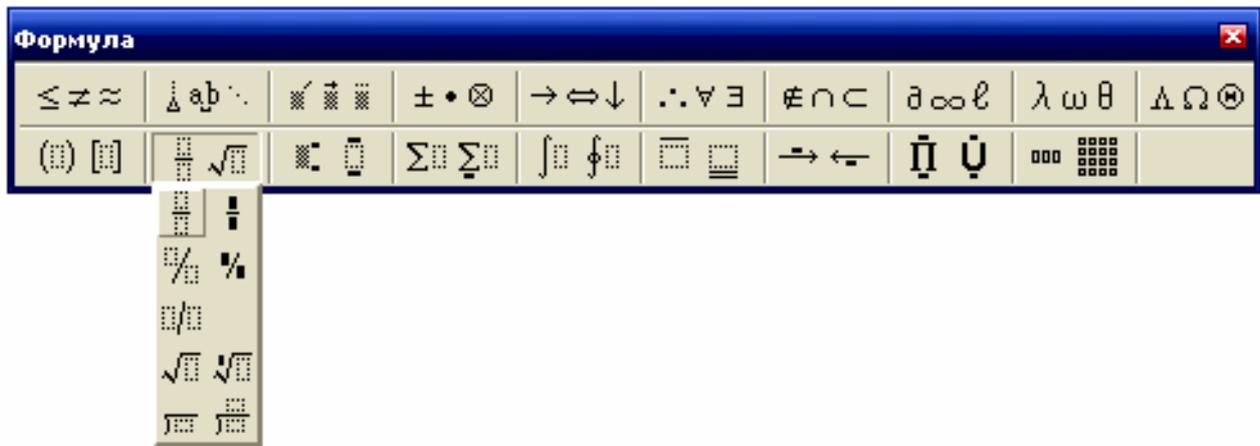


и вводя переменные и числа с клавиатуры. Верхняя строка панели инструментов **Формула** содержит более 150 математических символов. Нижняя строка используется для выбора разнообразных шаблонов, предназначенных для построения дробей, интегралов, сумм и других сложных выражений.

Чтобы поставить нижний индекс (при вводе  $Q_s$ ) используйте поле.



Чтобы поставить дробную черту используйте поле в нижней строке.



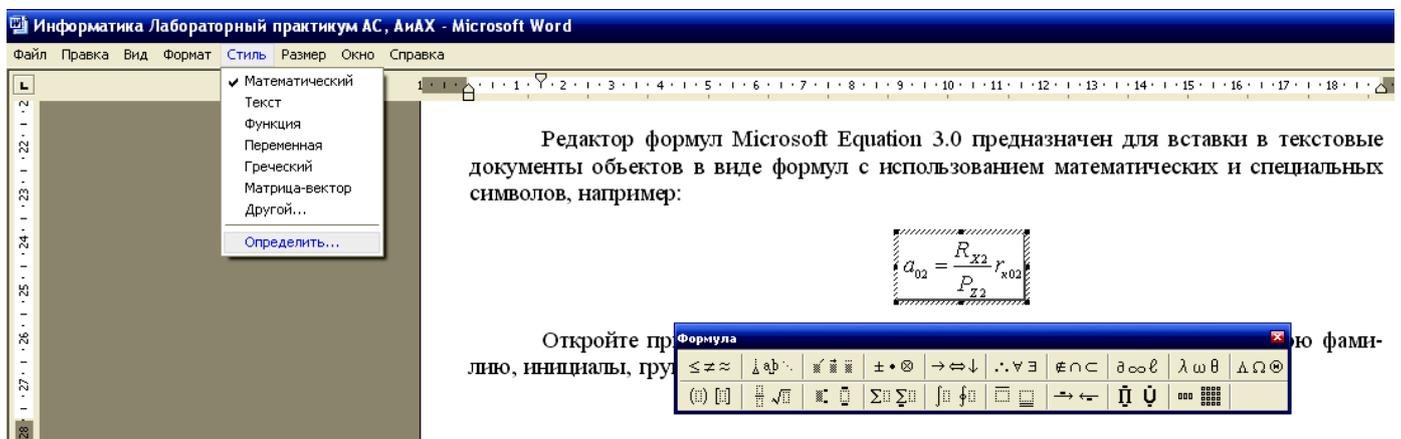
6. Чтобы вернуться в Microsoft Word, щелкните документ вне поля формулы.

7. Для повторного входа в формулу при необходимости ее редактирования активация производится двойным щелчком на поле формулы, либо нажатием правой клавишей на поле формулы и последующим выбором меню **Объект Формула**⇒**Открыть**.

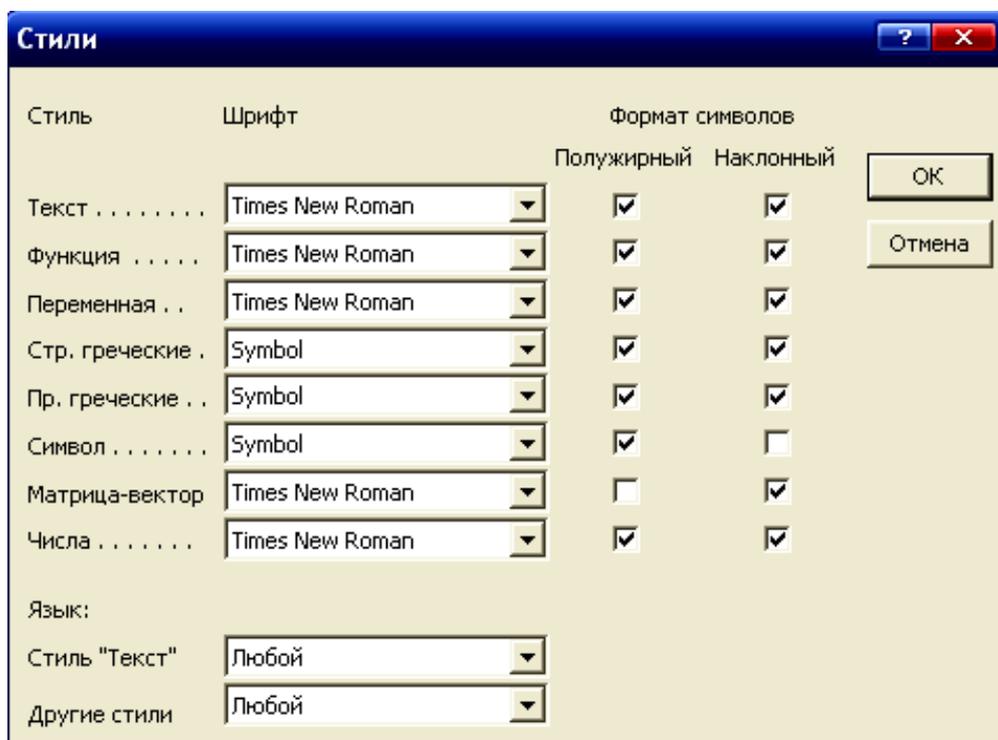
8. В редактор формул встроены элементы редактирования. Можно задавать размер и отображение всех символов, цифр, знаков, функций, текста.

Отредактируем параметры редактора формул, для этого войдите в ранее введенную формулу.

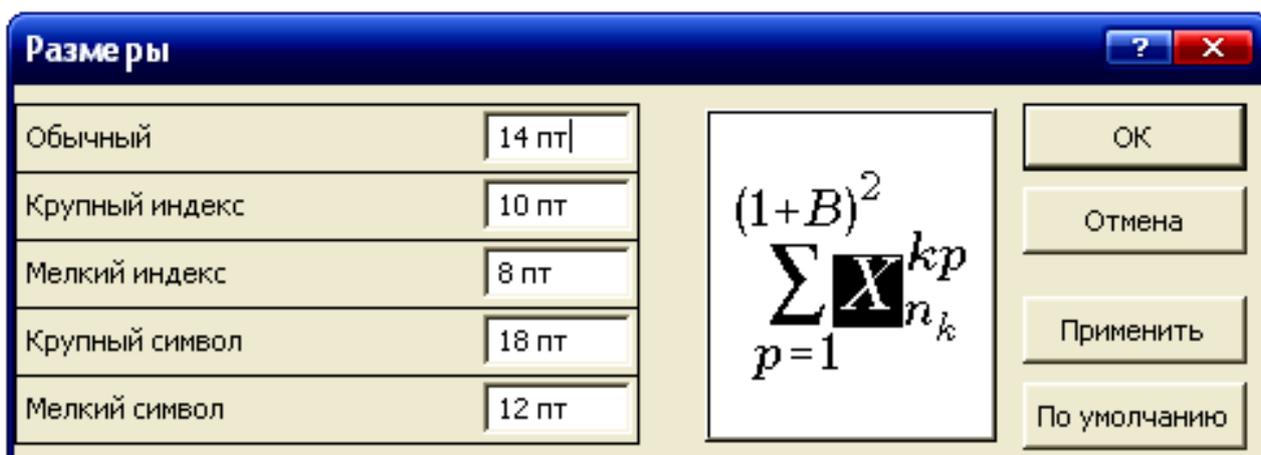
Откройте меню **Стиль**⇒**Определить**.



В открывшемся окне **Стили** установите параметры следующим образом:



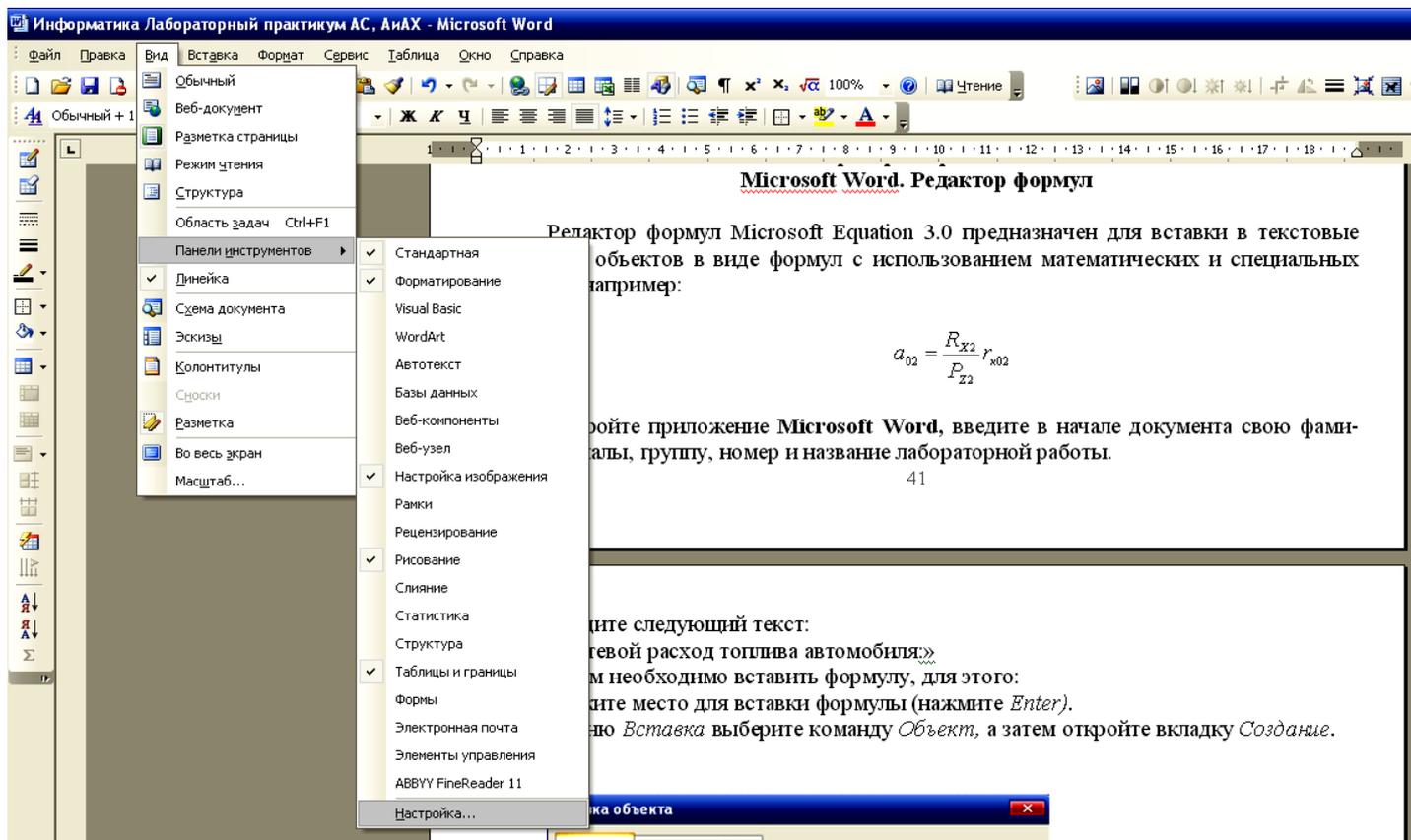
Откройте меню **Размер**⇒**Определить**. В открывшемся окне **Размеры** установите параметры следующим образом:



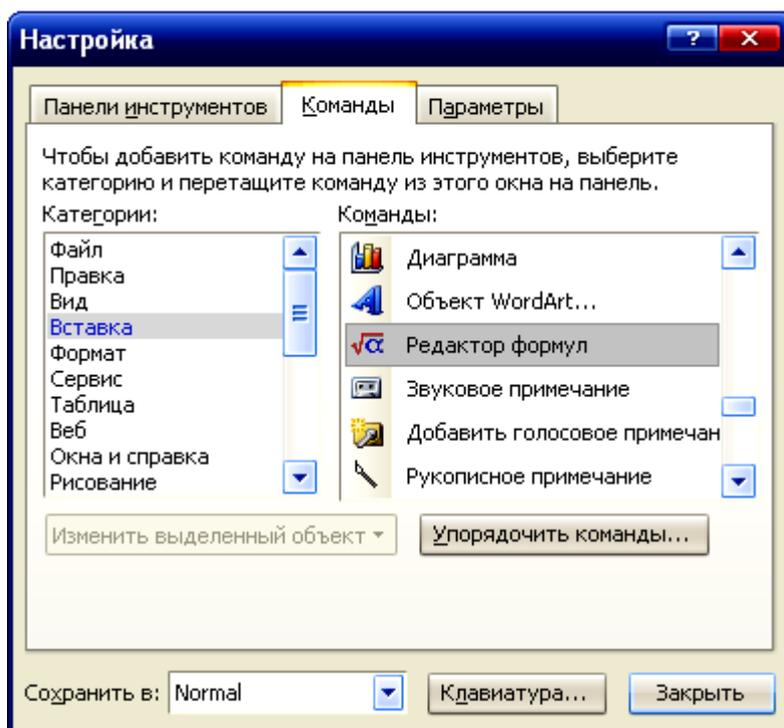
Вернитесь в Microsoft Word.

Вводить формулы удобнее, вызывая редактор формул клавишей  на панели инструментов. Если такая клавиша отсутствует на панели инструментов, то необходимо ее туда поместить.

Для этого откройте меню **Вид**⇒**Панели инструментов**⇒**Настройка**.



В открывшемся окне *Настройка* на вкладке *Команды* выбирается категория *Вставка*.



Необходимо найти команду



Нажимая и удерживая левую клавишу мыши на пиктограмме , перетащите ее на панель инструментов в нужное место и отпустите клавишу. Нажмите *Закреть*.

Введите все формулы и поясняющий текст, которые содержатся в образце далее. Используйте стили и форматирование как в образце.

### ОБРАЗЕЦ

Путевой расход топлива автомобиля определяется как:

$$Q_S = 100 \frac{dQ}{dS},$$

где  $Q_S$  – объем топлива, израсходованный автомобилем за 100 км пройденного пути, л/100 км;

$dQ$  – объем топлива, л;

$dS$  – путь, пройденный автомобилем, км.

Из теории автомобиля известно, что расход топлива находится по формуле:

$$Q_S = \frac{g_e (P_B + P_K + P_{II})}{36000 \rho_T \eta_{TP}}$$

где  $g_e$  – удельный эффективный расход топлива, г/кВт·ч;

$P_B$  – сила сопротивления воздуха, Н;

$P_K$  – сила сопротивления качению, Н;

$P_{II}$  – сила сопротивления подъему, Н;

$\rho_T$  – плотность топлива, кг/л;

$\eta_{TP}$  – коэффициент полезного действия трансмиссии.

За время  $dt$  автомобиль проходит путь:

$$dS = V_a dt$$

где  $V_a$  – скорость автомобиля, км/ч.

Для ТЭХ установившегося движения ( $V_a = const$ ) объем топлива, израсходованного за малый промежуток времени  $\Delta t$ , предлагается находить как:

$$\Delta Q = \frac{g_e (P_B + P_K + P_{II})}{12,96 \cdot 10^6 \rho_T \eta_{TP}} V_a \Delta t$$

где  $\Delta t$  – промежуток времени, с;

$\rho_T$  – плотность топлива, кг/м<sup>3</sup>.

Расход топлива  $Q_S$  на всем промежутке пути, пройденного за время  $t$ , предлагается определять как сумму расходов за каждый промежуток времени  $\Delta t$

$$Q_S = \sum \Delta Q.$$

Форма отчетности: отчет по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчет по лабораторному практикуму.

Основная литература [1,2].

Контрольные вопросы для самопроверки

1. Как открыть редактор формул?
2. Что такое панель редактора и какие её возможности?

## Лабораторная работа №6. Microsoft Excel. Основы работы.

Цель работы – получить навыки работы в программе *Microsoft Excel*, её интерфейса, научиться пользоваться основными средствами расчёта и редактирования документов.

Программа *Microsoft Excel* предназначена для работы с таблицами данных, преимущественно числовых. При формировании таблицы выполняют ввод, редактирование и форматирование текстовых и числовых данных, а также формул. Наличие средств автоматизации облегчает эти операции. Созданная таблица может быть выведена на печать.

Порядок выполнения:

### 1. Описание приемов работы

#### Рабочая книга и рабочий лист. Строки, столбцы, ячейки

Документ Excel называется рабочей книгой. Рабочая книга представляет собой набор рабочих листов, каждый из которых имеет табличную структуру и может содержать одну или несколько таблиц. В окне документа в программе Excel отображается только текущий рабочий лист, с которым и ведется работа. Каждый рабочий лист имеет название, которое отображается на ярлычке листа, отображаемом в его нижней части. С помощью ярлычков можно переключаться к другим рабочим листам, входящим в ту же самую рабочую книгу.

Рабочий лист состоит из *строк* и *столбцов*. Столбцы озаглавлены прописными латинскими буквами и, далее, двухбуквенными комбинациями. Всего рабочий лист может содержать до 256 столбцов, пронумерованных от A до IV. Строки последовательно нумеруются цифрами, от 1 до 65 536 (максимально допустимый номер строки).

На пересечении столбцов и строк образуются ячейки таблицы. Они являются минимальными элементами для хранения данных. Обозначение отдельной ячейки сочетает в себе номера столбца и строки (в этом порядке), на пересечении которых она расположена, например: A1 или DE234. Обозначение ячейки (ее номер) выполняет функции ее адреса. Адреса ячеек используются при записи формул, определяющих взаимосвязь между значениями, расположенными в разных ячейках. Одна из ячеек всегда является *активной* и выделяется *рамкой активной ячейки*. Эта рамка в программе Excel играет роль курсора. Операции ввода и редактирования всегда производятся в активной ячейке. Переместить рамку активной ячейки можно с помощью курсорных клавиш или указателя мыши. На данные, расположенные в соседних ячейках, можно ссылаться в формулах как на единое целое. Такую группу ячеек называют *диапазоном*. Наиболее часто используют прямоугольные диапазоны, образующиеся на пересечении группы последовательно идущих строк и группы последовательно идущих столбцов. Диапазон ячеек обозначают, указывая через двоеточие номера ячеек, расположенных в противоположных углах прямоугольника, например: A1:C15.

#### Ввод, редактирование и форматирование данных

Отдельная ячейка может содержать данные, относящиеся к одному из трех типов: *текст*, *число* или *формула*, а также оставаться пустой. Тип данных, размещаемых в ячейке, определяется автоматически при вводе. Если эти данные можно интерпретировать как число, программа Excel так и делает. В противном случае данные рассматриваются как текст. Ввод формулы всегда начинается с символа «=» (знака равенства).

Ввод текста и чисел. Ввод данных осуществляют непосредственно в текущую ячейку или в *строку формул*, располагающуюся в верхней части окна программы под панелями инструментов. Место ввода отмечается текстовым курсором. Если начать ввод нажатием алфавитно-цифровых клавиш, данные из текущей ячейки заменяются вводимым текстом. Если щелкнуть на строке формул или дважды на текущей ячейке, старое содержимое ячейки не удаляется и появляется возможность его редактирования. Вводимые данные в любом случае отображаются как в ячейке, так и в строке формул.

Чтобы завершить ввод, сохранив введенные данные, используют кнопку Ввод в строке формул или клавишу ENTER. Чтобы отменить внесенные изменения и восстановить

прежнее значение ячейки, используют кнопку Отмена в строке формул или клавишу ESC. Для очистки текущей ячейки или выделенного диапазона проще всего использовать клавишу DELETE.

### **Формулы и ссылки на ячейки**

Вычисления в таблицах программы Excel осуществляются при помощи **формул**. Формула может содержать числовые константы, ссылки на ячейки и **функции** Excel, соединенные знаками математических операций. Скобки позволяют изменять стандартный порядок выполнения действий. Если ячейка содержит формулу, то в рабочем листе отображается текущий результат вычисления этой формулы. Если сделать ячейку текущей, то сама формула отображается в строке формул.

Правило использования формул в программе Excel состоит в том, что, **если значение ячейки зависит от других ячеек таблицы, всегда следует использовать формулу, даже если операцию легко можно выполнить в «уме»**. Это гарантирует, что последующее редактирование таблицы не нарушит ее целостности и правильности производимых в ней вычислений.

Формула может содержать **ссылки**, то есть адреса ячеек, содержимое которых используется в вычислениях. Это означает, что результат вычисления формулы зависит от числа, находящегося в другой ячейке. Ячейка, содержащая формулу, таким образом, является **зависимой**. Значение, отображаемое в ячейке с формулой, пересчитывается при изменении значения ячейки, на которую указывает ссылка.

Ссылку на ячейку можно задать разными способами. Во-первых, адрес ячейки можно ввести вручную. Другой способ состоит в щелчке на нужной ячейке или выборе диапазона, адрес которого требуется ввести. Ячейка или диапазон при этом выделяются пунктирной рамкой.

Для редактирования формулы следует дважды щелкнуть на соответствующей ячейке. При этом ячейки (диапазоны), от которых зависит значение формулы, выделяются на рабочем листе цветными рамками, а сами ссылки отображаются в ячейке и в строке формул тем же цветом. Это облегчает редактирование и проверку правильности формул.

По умолчанию, ссылки на ячейки в формулах рассматриваются как **относительные**. Это означает, что при копировании формулы адреса в ссылках автоматически изменяются в соответствии с относительным расположением исходной ячейки и создаваемой копии.

При **абсолютной адресации** адреса ссылок при копировании не изменяются, так что ячейка, на которую указывает ссылка, рассматривается как не табличная. Для изменения способа адресации при редактировании формулы надо выделить ссылку на ячейку и нажать клавишу F4. Элементы номера ячейки, использующие абсолютную адресацию, предваряются символом \$. Например, при последовательных нажатиях клавиши F4 номер ячейки A1 будет записываться как A1, \$A\$1, A\$1 и \$A1. В двух последних случаях один из компонентов номера ячейки рассматривается как абсолютный, а другой – как относительный.

### **Копирование содержимого ячеек**

Копирование и перемещение ячеек в программе Excel можно осуществлять методом перетаскивания или через буфер обмена. Чтобы методом перетаскивания скопировать или переместить текущую ячейку (выделенный диапазон) вместе с содержимым, следует навести указатель мыши на рамку текущей ячейки (он примет вид стрелки с дополнительными стрелочками). Теперь ячейку можно перетащить в любое место рабочего листа (точка вставки помечается всплывающей подсказкой). Для выбора способа выполнения этой операции, а также для более надежного контроля над ней рекомендуется использовать **специальное перетаскивание** с помощью правой кнопки мыши. В этом случае при отпускании кнопки мыши появляется контекстное меню, в котором можно выбрать конкретную выполняемую операцию.

Передача информации через буфер обмена имеет в программе Excel определенные особенности. Вначале необходимо выделить копируемый (вырезаемый) диапазон и дать команду на его помещение в буфер обмена (копировать или вырезать). Вставка данных в рабо-

чий лист возможна лишь немедленно после их помещения в буфер обмена. Попытка выполнить любую другую операцию приводит к отмене начатого процесса копирования или перемещения. Место вставки определяется путем указания ячейки, соответствующей верхнему левому углу диапазона, помещенного в буфер обмена, или путем выделения диапазона, который по размерам в точности равен копируемому (перемещаемому). Для управления способом вставки можно использовать команду **Специальная вставка**. В этом случае правила вставки данных из буфера обмена задаются в открывшемся диалоговом окне.

### Использование стандартных функций

Стандартные функции используются в программе Excel только в формулах. **Вызов функции** состоит в указании в формуле *имени функции*, после которого в скобках указывается *список параметров*. Отдельные параметры разделяются в списке точкой с запятой. В качестве параметра может использоваться число, адрес ячейки или произвольное выражение, для вычисления которого также могут использоваться функции.

В режиме ввода формулы в левой части строки формул, где раньше располагался номер текущей ячейки, появляется раскрывающийся список функций. Он содержит десять последних функций, а также пункт **Другие функции**. Как только имя функции выбрано, на экране появляется диалоговое окно **Аргументы функции**. Это окно, в частности, содержит значение, которое получится, если немедленно закончить ввод формулы. Правила вычисления формул, содержащих функции, не отличаются от правил вычисления более простых формул.

## 2. Задание на работу с таблицами

1) Запустите программу Excel (**Пуск** → **Программы** → **Microsoft Office** → **Excel**). Переименуйте рабочий лист1 на «Данные», дважды щелкнув на его ярлычке. Сохраните файл в своей папке под именем Электронные таблицы.xls.

2) Сделайте активной ячейку A1. Введите текст «Расчет стоимости закупки». Увеличьте ширину столбца A по ширине введенного текста. В ячейку A3 введите текст «Величина скидки, %». В ячейку C3 введите 1,5. В ячейку A5 введите текст «Прайс-лист».

3) Начиная с ячейки A7 ввести данные из таблицы:

Наименование товара:	Количество	Цена, руб	Стоимость, руб	Доставка, руб	К оплате, руб
Шина 205/70 R14	16	600		70	
Диск сцепления ГАЗ-3110	2	850,5		0	
Двигатель ЗМЗ-406	5	56000		2500	
Рычаг КП	1	253,5		0	
Рессора ГАЗ	2	670		100	
Рулевой механизм ГАЗ	1	2458,85		0	
Свеча зажигания А-14Д	24	35		0	
Фильтр масляный	5	260		0	
Фильтр топливный	7	150		0	
Фильтр воздушный	8	120		0	
Накладки торм. ГАЗ	4	65		0	
Зеркало боковое, лев	1	1650,2		0	
Масло мот. SAE 20/W40	50	24,5		120	
<b>Итого, руб:</b>					
<b>Сумма со скидкой, руб:</b>					

Подберите ширину ячеек. Выделите диапазон A7-F7, нажмите правой клавишей мыши на диапазоне, в контекстном меню выбрать «Формат ячеек». Вкладка «Выравнива-

ние»: по горизонтали и вертикали выбрать «По центру». Вкладка «Вид»: выбрать заливку серого цвета.

4) Выделите диапазон A7:F23, нажмите правой клавишей мыши на диапазоне, в контекстном меню выбрать «Формат ячеек». Вкладка «Граница»: нажать на «внешние» и «внутренние».

5) Вычислим стоимость товара, перемножив количество на цену. Перейдите в ячейку D8. Через знак равенства вводим формулу =B8\*C8, нажать ENTER. Выделите ячейку D8. Наведите указатель мыши на маркер заполнения в правом нижнем углу рамки ячейки, нажмите и удерживайте левую клавишу мыши, перетащите маркер, захватив диапазон, включая ячейку D20.

6) Сложите стоимость товара со стоимостью доставки, тем самым, определите сумму к оплате. Перейдите в ячейку F8. Через знак равенства вводим формулу =D8+E8, нажать ENTER. Выделите ячейку F8. Наведите указатель мыши на маркер заполнения в правом нижнем углу рамки ячейки, нажмите и удерживайте левую клавишу мыши, перетащите маркер, захватив диапазон, включая ячейку F20.

7) Подведите итоговую сумму. Перейдите в ячейку F22. С помощью кнопки «Автосумма»  на панели инструментов вставляем в ячейку функцию СУММ( ) и выделяем протягиванием мыши диапазон ячеек F8:F20, по которому будет производиться суммирование. Формула должна выглядеть следующим образом: =СУММ(F8:F20). Нажмите ENTER.

8) Определите сумму с учетом предоставленной скидки. Перейдите в ячейку F23. Через знак равенства вводим формулу =F22-F22\*C3%, нажать ENTER.

### 3. Построение диаграмм и графиков

В программе Excel термин «диаграмма» используется для обозначения всех видов графического представления числовых данных. Построение графического изображения производится на *основе ряда данных*. Так называют группу ячеек с данными в пределах отдельной строки или столбца. На одной диаграмме можно отображать несколько рядов данных.

Диаграмма представляет собой вставной объект, внедренный на один из листов рабочей книги. Она может располагаться на том же листе, на котором находятся данные, или на любом другом листе (часто для отображения диаграммы отводят отдельный лист). Диаграмма сохраняет связь с данными, на основе которых она построена, и при обновлении этих данных немедленно изменяет свой вид.

Для построения диаграммы обычно используют Мастер диаграмм, запускаемый щелчком на кнопке Мастер диаграмм на стандартной панели инструментов. Часто удобно заранее выделить область, содержащую данные, которые будут отображаться на диаграмме, но задать эту информацию можно и в ходе работы мастера.

На первом этапе работы мастера выбирают форму диаграммы. Доступные формы перечислены в списке **Тип** на вкладке **Стандартные**. Для выбранного типа диаграммы справа указывается несколько вариантов представления данных (палитра **Вид**), из которых следует выбрать наиболее подходящий. На вкладке **Нестандартные** отображается набор полностью сформированных типов диаграмм с готовым форматированием. После задания формы диаграммы следует щелкнуть на кнопке **Далее**.

Второй этап работы мастера служит для выбора данных, по которым будет строиться диаграмма. Если диапазон данных был выбран заранее, то в области предварительного просмотра в верхней части окна мастера появится приблизительное отображение будущей диаграммы. Если данные образуют единый прямоугольный диапазон, то их удобно выбирать при помощи вкладки **Диапазон данных**. Если данные не образуют единой группы, то информацию для отрисовки отдельных рядов данных задают на вкладке **Ряд**. Предварительное представление диаграммы автоматически обновляется при изменении набора отображаемых данных.

Третий этап работы мастера (после щелчка на кнопке Далее) состоит в выборе оформления диаграммы. На вкладках окна мастера задаются:

- название диаграммы, подписи осей (вкладка Заголовки);
- отображение и маркировка осей координат (вкладка Оси);
- отображение сетки линий, параллельных осям координат (вкладка Линии сетки);
- описание построенных графиков (вкладка Легенда);
- отображение надписей, соответствующих отдельным элементам данных на графике (вкладка Подписи данных);
- представление данных, использованных при построении графика, в виде таблицы (вкладка Таблица данных).

На последнем этапе работы мастера (после щелчка на кнопке Далее) указывается, следует ли использовать для размещения диаграммы новый рабочий лист или один из имеющихся. Обычно этот выбор важен только для последующей печати документа, содержащего диаграмму. После щелчка на кнопке Готово диаграмма построится автоматически и будет вставлена на указанный рабочий лист.

Готовую диаграмму можно изменить. Она состоит из набора отдельных элементов, таких, как сами графики (ряды данных), оси координат, заголовок диаграммы, область построения и прочее. При щелчке на элементе диаграммы он выделяется маркерами, а при наведении на него указателя мыши — описывается всплывающей подсказкой. Открыть диалоговое окно для форматирования элемента диаграммы можно через меню Формат (для выделенного элемента) или через контекстное меню (команда Формат). Различные вкладки открывшегося диалогового окна позволяют изменять параметры отображения выбранного элемента данных.

#### 4. Задание на построение диаграмм

1) Перейдите в Лист2 документа переименуйте его в «Диаграмма».

2) В ячейку A1 введите текст «Коэффициент технической готовности парка за год». В ячейку A3, введите «Месяц года», в ячейку A4 – «январь». Заполните остальные ячейки по месяцам, используя автозаполнение: подведите курсор в нижний правый угол ячейки A4 до появления черного перекрестия, нажмите левую клавишу мыши и удерживая ее передвигайте курсор вниз, до появления подсказки с месяцем декабрь. Начиная с ячейки B3, введите столбец данных:

Кгт
0,85
0,95
0,75
0,7
0,65
0,73
0,78
0,83
0,86
0,9
0,95
0,8

3) Выделите диапазон A4:B15. Запустите Мастер диаграмм нажатием на клавишу



В открывшемся окне выбрать на вкладке Стандартные гистограмму объемную (первый ряд, вторая строка в таблице Вид). Нажать Далее. Выбрать «ряды в столбцах», нажать Далее, Добавить легенду – убрать галочку. На вкладке «Линии сетки» установить основные линии

на оси X. На вкладке «Подписи данных» поставить галочку «значения». Нажать Далее, поместить диаграмму на имеющемся листе, нажать Готово.

4) Навестись на любой ряд диаграммы до появления подсказки, нажать правой клавишей и выбрать «Формат ряда данных». На вкладке Фигура выбрать цилиндрическую форму, на вкладке Параметры выбрать ширину зазора 50, нажать ОК.

5) Навестись на подпись данных первого ряда (0,85) диаграммы до появления подсказки, нажать правой клавишей и выбрать «Формат подписей данных». Выбрать на вкладке «Шрифт» начертание полужирный, размер 8пт.

6) Навестись на фон диаграммы до появления подсказки «стенки», нажать правой клавишей и выбрать «Объемный вид». Установить возвышение 30, поворот 30.

7) Навестись на ось категорий (на слово январь), нажать правой клавишей и выбрать «Формат оси». На вкладке выравнивание установить ориентацию 90.

8) Сохранит файл.

9) Перейдите в Лист3 документа переименуйте его в «График».

10) В ячейку A1 введите текст «Тягово-скоростная характеристика автомобиля». Начиная с ячейки A3, введите данные как в таблице:

Va	Pт	Pд+Pв
40	9778	3238
52,5	10451	3900
65	10945	4449
77,5	11260	5116
90	11397	5899
102,5	11356	6799
114,9	11135	7816
127,4	10736	8950
140	10159	10201

11) Выделите диапазон A4:C12. Запустите Мастер диаграмм нажатием на клавишу



В открывшемся окне выбрать на вкладке Стандартные точечную диаграмму со значениями, соединенными сглаживающими линиями (первый ряд, вторая строка в таблице Вид). Нажать Далее. Выбрать «ряды в столбцах», нажать Далее. На вкладке «Легенда» Добавить легенду – убрать галочку. На вкладке «Заголовки»: Название диаграммы – «ТСХ», Ось X – «Скорость», Ось Y – «Тяговая сила». На вкладке «Линии сетки»: установить промежуточные линии оси X. Нажать Далее, поместить диаграмму на имеющемся листе, нажать Готово.

Форма отчетности: отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

Основная литература [1,2].

Контрольные вопросы для самопроверки

1. Что такое автоматизированные расчёты?
2. Что такое рабочее поле?

### **Лабораторная работа №7. Использование ссылок.**

Цель работы – научиться использовать автоматически изменяющиеся ссылки в Microsoft Excel.

Порядок выполнения:

Задание 1 :

При составлении формул используются так называемые, относительные ссылки на ячейки. Различие между этими ссылками с другими проявляются при распространении и копировании формулы в другие ячейки. При изменении положения формулы, автоматически меняются ссылки на ячейки относительно исходной.

Воспользуйтесь справочной системой MS Excel и найдите описание относительной ссылки, самостоятельно разберите основные принципы создания относительных ссылок.

Подготовьте таблицу по образцу:

Наименование товара	Количество			
	1	2	3	4

Введите наименование товара и цену за единицу:

Наименование товара	Количество			
	1	2	3	4
SIMM 4MB 72PIN	55,80			

Для того чтобы рассчитать стоимость товара за две единицы, выделите соответствующую ячейку (ниже ячейки с цифрой 2) и в строке формул введите название ячейки со стоимостью товара за 1 шт., в нашем случае, это ячейка в которой содержится число 55,80. После чего, ставим знак умножить \* и пишем название ячейки, определяющей количество товара, т.е. ячейку со значением 2. Остальные ячейки заполните самостоятельно. Распространите формулу вправо (используя маркер заполнения) чтобы получить стоимость товара за 3 и 4 единицы. Можно заметить, что вычисление по формуле стоимости товара за три единицы неверно.

Наименование товара	Количество			
	1	2	3	4
SIMM 4MB 72PIN	55,80	111,60	334,80	

В таких случаях, составляя формулу, применяют, абсолютные ссылки. При перемещении или копировании формулы абсолютные ссылки не изменяются, в то время как относительные ссылки автоматически обновляются в зависимости от нового положения. Абсолютные ссылки имеют вид \$F\$9, \$G\$5 и т.д. Для того чтобы в формулу ввести абсолютную ссылку необходимо ячейку с 55.80 записать со знаком \$.

Наименование товара	Количество			
	1	2	3	4
SIMM 4MB 72PIN	55,80	111,60	167,40	223,20

Таким образом, получим правильное решение формул при копировании. Самостоятельно с помощью Справки MS Excel изучите понятия ссылок (их программирование). Довольно часто применяются не чисто относительные или абсолютные ссылки, а смешанные

ссылки, например C\$1 или \$C1. Часть ссылки, не содержащая знака \$, будет обновляться при копировании, а другая часть, со знаком \$, останется без изменения. В первом случае будет фиксированным положение строки (при перемещении формулы, данные будут браться только из первой строки), а во втором случае, зафиксирован столбец, т.е. Ссылка относится к столбцу C, а положение строки изменяется относительно перемещения формулы.

**Задание2:**

Составьте таблицу степеней для первого десятка. Для этого заготовьте таблицу по образцу. По вертикали расположены числа, а по горизонтали показатели степеней.

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						
7						
8						
9						

В первую ячейку (вычисление 1x1) введите первую формулу для вычисления степени. Для этого воспользуйтесь Мастером функций (**Вставка** → **Функция**),

или кнопкой  панели инструментов, или после ввода знака = в раскрывающемся списке Поля имени равенства, выберите функцию

- 1) Выберите функцию **СТЕПЕНЬ**.
- 2) Введите ссылки для числа и показатели степени.

Попробуйте распространить эту формулу вниз. В результате получите следующую картину:

	1	2	3	4	5	6
1	1					
2	2					
3	9					
4	262144					
5	#ЧИСЛО!					
6	#ЧИСЛО!					
7	#ЧИСЛО!					
8	#ЧИСЛО!					
9	#ЧИСЛО!					

По каким-то причинам результат вычисляется неверно. Можно заметить, что числа необходимо брать только из столбца с цифрами от 1 до 9, т.е. в исходной формуле необходимо зафиксировать столбец с числами и строку со степенью. После чего можно производить распространение формулы в соседние ячейки. Ниже приведена таблица с правиль-

ным вводом формулы. Соответственно вы можете проверить правильность расчета формулы в ячейках:

		Степень					
		1	2	3	4	5	6
	1	=СТЕПЕНЬ(\$O35;P\$34)					
	2	1	4	8	16	32	64
	3	1	9	27	81	243	729
	4	1	16	64	256	1024	4096
	5	1	25	125	625	3125	15625
	6	1	36	216	1296	7776	46656
	7	1	49	343	2401	16807	117649
	8	1	64	512	4096	32768	262144
	9	1	81	729	6561	59049	531441

**Задание 3:**

Составьте таблицу кубов для чисел первого десятка.

		Десятки								
		1	2	3	4	5	6	7	8	9
<b>Единицы</b>	1	1331	9261	29791	68921	132651	226981	357911	531441	753571
	2	1728	10648	32768	74088	140608	238328	373248	551368	778688
	3	2197	12167	35937	79507	148877	250047	389017	571787	804357
	4	2744	13824	39304	85184	157464	262144	405224	592704	830584
	5	3375	15625	42875	91125	166375	274625	421875	614125	857375
	6	4096	17576	46656	97336	175616	287496	438976	636056	884736
	7	4913	19683	50653	103823	185193	300763	456533	658503	912673
	8	5832	21952	54872	110592	195112	314432	474552	681472	941192
	9	6859	24389	59319	117649	205379	328509	493039	704969	970299

Запишите формулу, которую использовали при вычислении кубов.

**Задание 4:**

Создайте таблицу по образцу.

<b>Зарботная плата</b>		4000
<b>Виды начислений</b>	<b>В процентах к зарплате</b>	<b>Размер начислений</b>
На пенсионное обеспечение	13,00	520,00
На медицинское страхование	8,43	337,20
На народную пенсию (для обеспечения тех, у кого нет выслуги лет)	5,86	234,40
На пенсионную страховку	0,20	8,00
На страхование от производственного травматизма	1,38	55,20
На превентивную защиту сотрудников	0,17	6,80
На выплату пособий по безработице, финансирование переквалификации потерявших работу	2,12	84,80
На гарантию зарплаты при банкротстве компании	0,20	8,00
На страхование жизни	0,61	24,40
На групповую страховку	0,95	38,00
На пособия при увольнении	0,28	11,20
На дополнительную пенсию	3,10	124,00
На страховку от травматизма	1,00	40,00
В гарантийный фонд	0,06	2,40
Налог на пенсионные платежи	0,60	24,00

Введите формулу для вычисления размера начислений. Введите значение заработной платы.

При изменении заработной платы - автоматически изменятся размеры начислений. Социальные платежи в виде начислений на фонд заработной платы труда - одна из крупных статей доходов в бюджетах западных государств. В таблице приведен пример начислений на заработную плату работников наемного труда в Швеции.

На примере приведены варианты формулы по вычислению размера начислений

<b>Зарботная плата</b>		100
	<b>В процентах к зарплате</b>	<b>Размер начислений</b>
	13,00	=J\$77*(H\$81)%
	8,43	8,43

<b>Зарботная плата</b>		100
	<b>В процентах к зарплате</b>	<b>Размер начислений</b>
	13,00	13,00
	8,43	=J\$77*H\$82/100
я тех, у	5,86	5,86

**Форма отчетности:** отчет по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчет по лабораторному практикуму.

**Основная литература** [1,2].

**Контрольные вопросы для самопроверки**

1. Что такое папка?
2. Что такое панель задач?

**Лабораторная работа №8.** Microsoft Excel. Построение диаграмм.

Цель работы – научиться строить и форматировать диаграммы (графики) в Excel, просматривать и распечатывать их на принтере.

Порядок выполнения

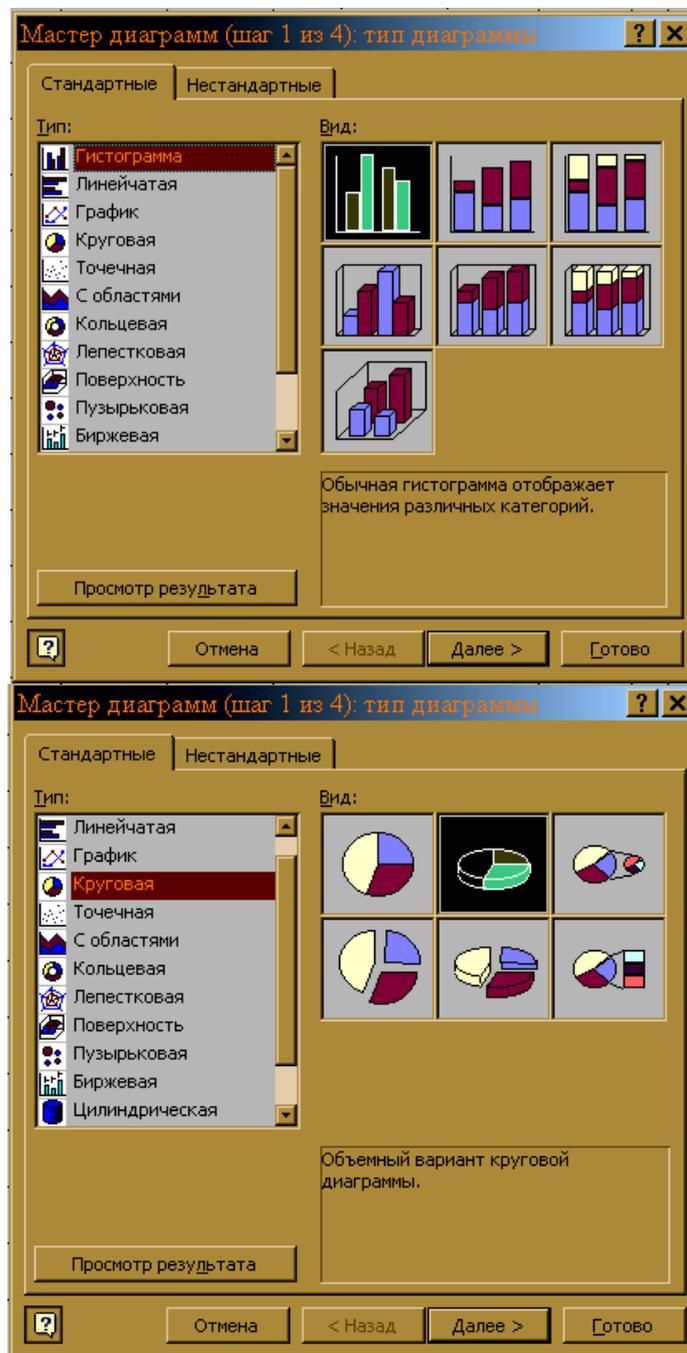
**Задание 1:**

Подготовьте таблицу по образцу.

Административный округ	Численность рабочих
Центральный	1194,7
Северный	512,8
Северо-Западный	196,6
Северо-Восточный	353,2
Южный	438,9
Юго-Западный	272,1
Юго-Восточный	373,8
Западный	366,4
Восточный	427,8
г. Зеленоград	77,5

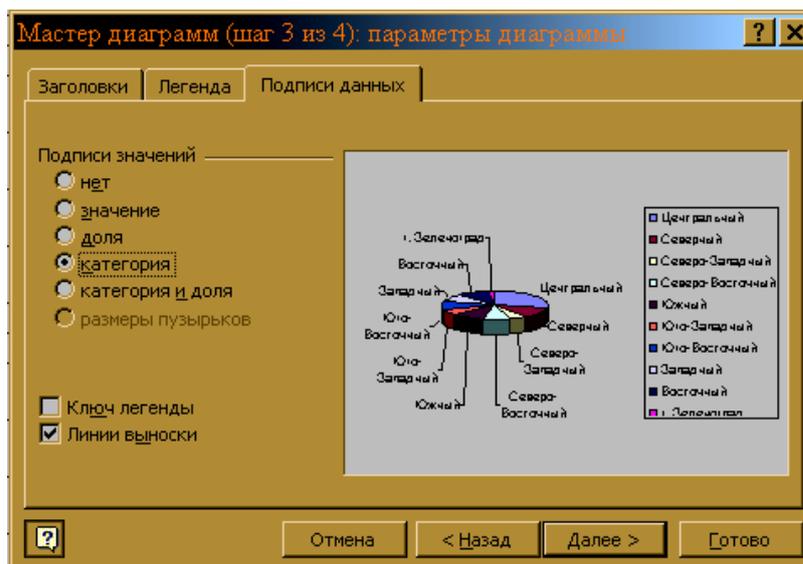
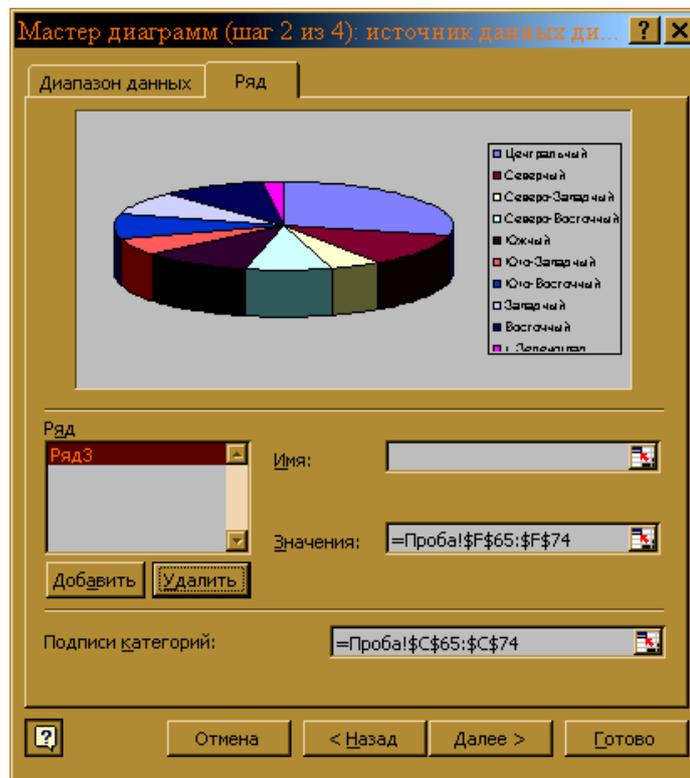
Выделите таблицу со строкой заголовка.

В меню **Вставка** выберите команду **Диаграмма**. Или выберите кнопку **Мастер Диаграмм** . Начнет работать Мастер Диаграмм.



В первом окне **Мастера Диаграмм** выберите тип диаграммы – круговую объемную. Кнопка **Просмотр** результата позволяет увидеть диаграмму. Нажмите кнопку **Далее**. В следующем окне отображается выделенный диапазон ячеек. Нажмите кнопку **Далее**.

На следующем шаге, выбирая вкладки диалогового окна можно корректировать название диаграммы. На вкладке **Подписи** значений выберите **Категория** и нажмите **Далее**.

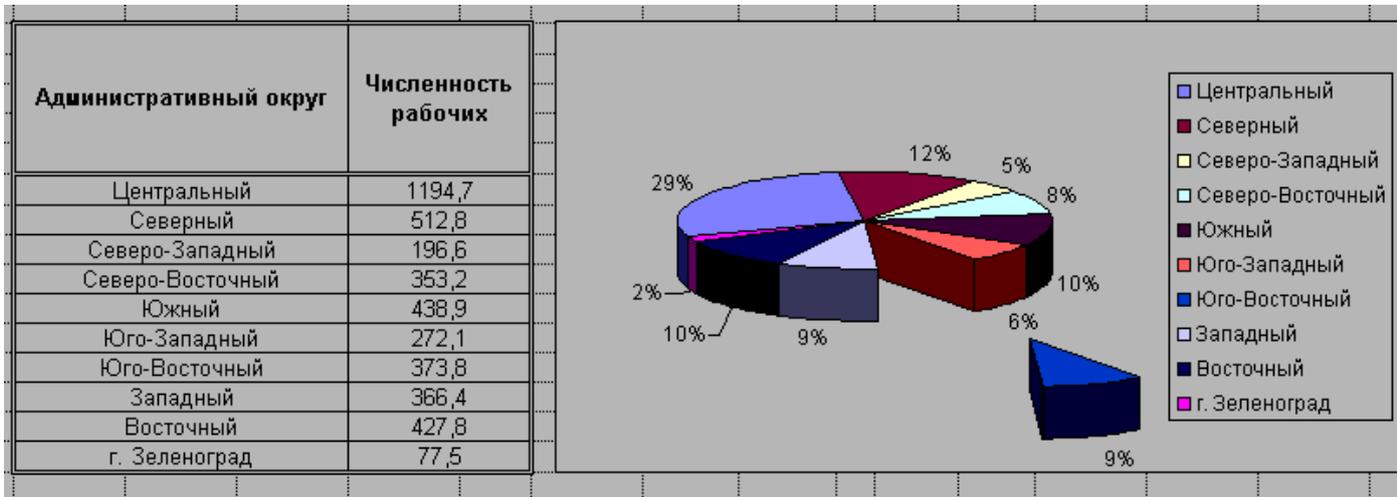


На следующем шаге определите положение диаграммы и выберите кнопку **Готово**. Диаграмма построена. На экране одновременно должны быть видны и таблица, и диаграмма. В случае необходимости диаграмму можно перенести и изменить ее пропорции (растянуть или сжать). Одиночным щелчком выделите область диаграммы. Поочередно выберите пункты горизонтального меню **Вставка** и **Формат** и обратите внимание на изменение команд, а также вид рамки вокруг диаграммы. Вы находитесь в режиме редактирования диаграммы и можете ее изменить.

Самостоятельно с помощью Справки MS Excel разобратъ свойства диаграмм, их параметры и тип.

### **Задание 2:**

На основе созданной диаграммы выберите команду **Объемный вид** и поверните диаграмму таким образом, чтобы подписи располагались наиболее оптимально.



Так как показано на примере отделите Юго-Восточный округ от общей круговой диаграммы и покажите на диаграмме процентное соотношение каждой доли.

Изменить цвет доли Северо-Западного округа на Зеленый.

### Задание 3:

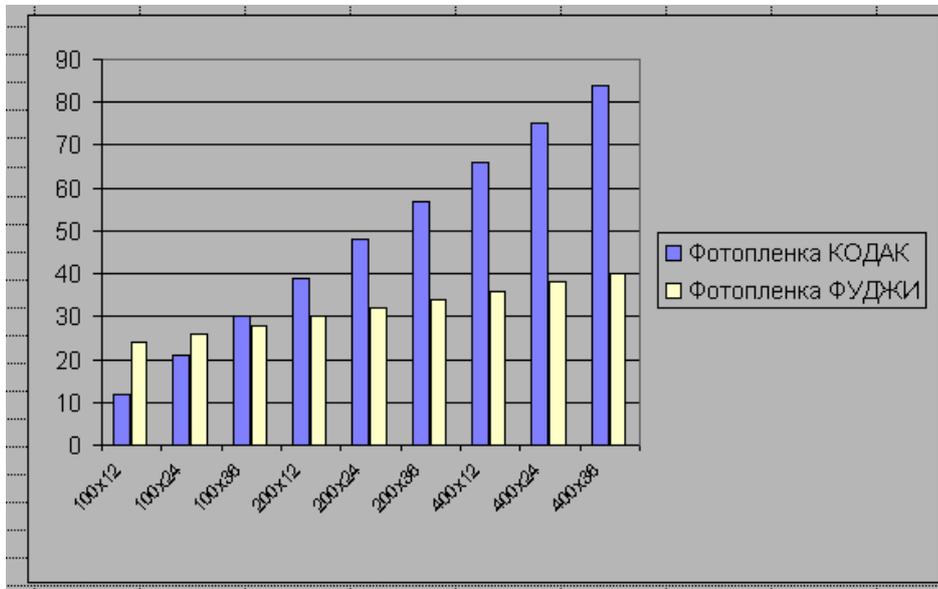
Представьте себя менеджером фирмы по продаже фотопленки. Ежедневно вы подводите итоги продаж и планируете объем заказа на складе. Подготовьте по приведенному ниже образцу и заполните ее по своему усмотрению (внесите количество проданных пленок каждого вида)

Объем продаж на	
" "	200 г.
	Фотопленка КОДАК
100x12	
100x24	
100x36	
200x12	
200x24	
200x36	
400x12	
400x24	
400x36	

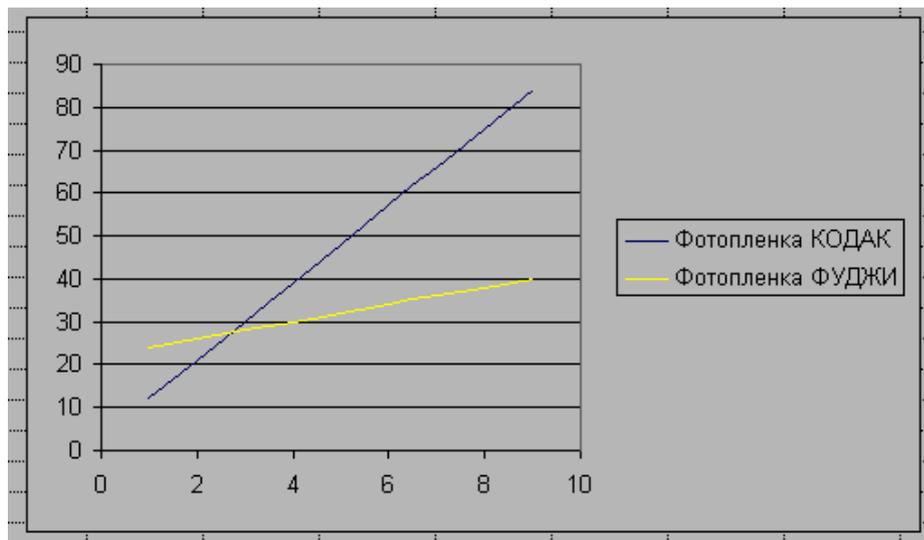
Объем продаж на		
" "	200 г.	
	Фотопленка КОДАК	Фотопленка ФУДЖИ
100x12		
100x24		
100x36		
200x12		
200x24		
200x36		
400x12		
400x24		
400x36		

Выделите свою таблицу и запустите **Мастер Диаграмм**. Выберите тип диаграммы – **Объемная гистограмма**. Далее в процессе формирования диаграммы обратите внимание на изменение вкладок диалогового окна **Параметры диаграммы**. Это вызвано изменением типа диаграммы. Самостоятельно рассмотрите форматирование оси, выбор заливки для различных элементов диаграммы, добавление нового ряда значений.

После того как вы построили первую гистограмму, добавьте второй столбец с названием Фотопленка ФУДЖИ и произвольно заполните его.



На пример гистограммы отчетливо виден объем продаж той или иной пленки. На базе уже созданной таблицы постройте линейный график объема продаж пленки (см. пример ниже), а также измените тип линии для фотопленки Кодак и ее цвет.

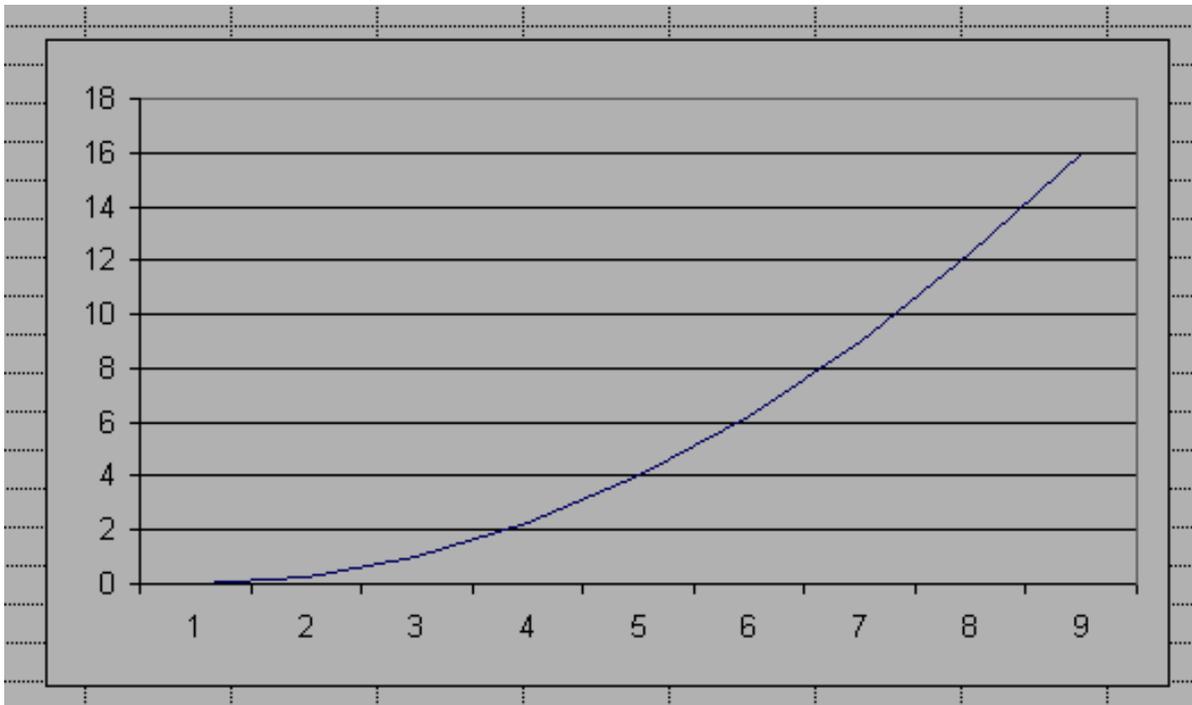


#### Задание 4:

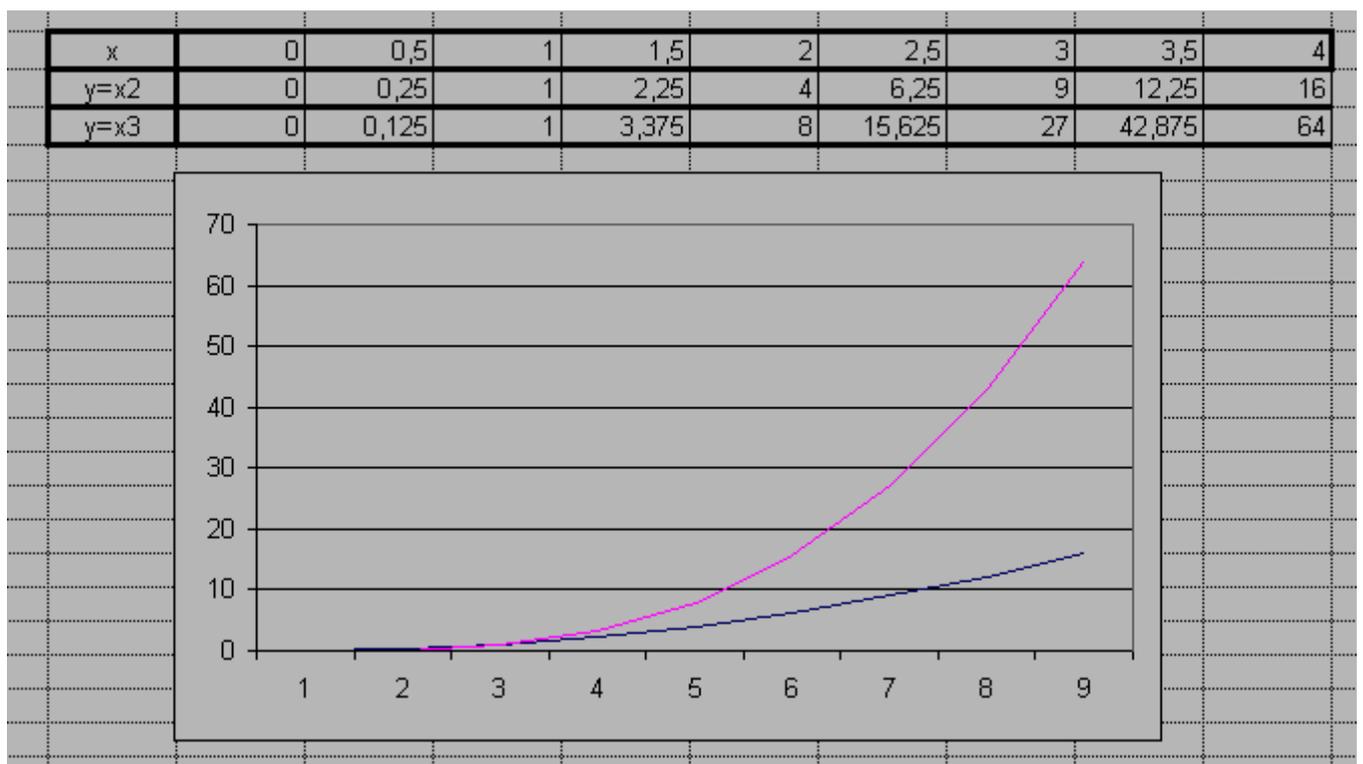
Воспользовавшись мастером функций, составьте таблицу значений функций  $y = x^2$ , для значений аргумента от 0 до 4 с шагом 0.5. Для заполнения ряда абсцисс примените маркер заполнения. Подберите ширину столбцов.

x	0	0,5	1	1,5	2	2,5	3	3,5	4
$y=x^2$									

Введите формулу и вычислите значения функции. Выделите только значения функции Y, в противном случае у вас получится два графика (по данным первой строки график линейной функции и по данным второй строки – график квадратичной функции). Запустить мастер диаграмм и выбрать тип диаграммы – **График**. Постройте график, следуя указаниям **Мастера диаграмм**.



К исходной таблице добавьте новый ряд данных для значений функции  $y = x^3$ .



Щелчком в области диаграммы перейдите в режим диаграммы. Добавьте к графику новые данные, выполнив команду (**Диаграмма**→**Добавить данные**) и выделите в таблице добавляемые данные. Аналогично к исходной таблице и графику добавьте еще два ряда значений для функции  $y = x^4$  и  $y = x^5$ .

Форма отчетности: отчет по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчет по лабораторному практикуму.

Основная литература [1,2].

Контрольные вопросы для самопроверки

1. Как строить график в Excel?

2. Как выполнять несколько осей координат с различными величинами по вертикали?
3. Какими средствами изменять толщину линий графика и размер точек?

### **Лабораторная работа №9. Microsoft Power Point. Создание презентаций.**

Цель работы – научиться создавать программу с наглядной иллюстрацией материала доклада и транслировать его на экран во время доклада с помощью компьютерных средств.

#### Порядок выполнения.

**1. Презентация** – это представление информации для некоторой целевой аудитории, с использованием разнообразных средств привлечения внимания и изложения материала.

Очень важен выбор оптимального объема презентации, он зависит от цели, для которой создается презентация, от предполагаемого способа ее использования.

#### **Логическая последовательность создания презентации (этапы подготовки):**

1. Структуризация материала.
2. Составление сценария реализации материала.
3. Разработка дизайна презентации.
4. Подготовка медиа-фрагментов (тексты, иллюстрации, аудио-фрагменты, видеофрагменты, анимация).
5. Тестирование-проверка, доводка презентации.

#### **2. Структуризация материала**

На основе литературы отбирается необходимая содержательная часть, формулируются основные тезисы, определяются ключевые моменты и ключевые слова, то есть выстраивается концепция – прежде чем приступить к работе над презентацией, следует добиться полного понимания того, о чем вы собираетесь рассказывать.

#### **Составление сценария реализации материала**

Презентация со сценарием – показ слайдов под управлением ведущего. Такие презентации могут содержать "плывущие" по экрану титры, анимированный текст, диаграммы, графики и другие иллюстрации. При этом автор должен понимать, что объекты, особенно меняющие положение, должны быть обоснованы целью презентации. Сами по себе это объекты отвлекают внимание и могут мешать воспринимать информацию. Порядок смены слайдов, а также время демонстрации каждого слайда определяет докладчик. Он же произносит текст, комментирующий видеоряд презентации.

Работу стоит **начинать с оставления плана** будущей презентации. Желательно, чтобы план был подробным. Необходимо на бумаге нарисовать структуру презентации, схематическое изображение слайдов и прикинуть, какой текст, рисунки, фотографии или другие материалы будут включены в тот или другой слайд. Составляется список рисунков, фотографий, звуковых файлов, видеороликов (если они необходимы), которые будут размещены в презентации. Определяется текстовая часть презентации.

При создании презентации необходимо найти правильный **баланс** между подаваемым материалом и сопровождающими его мультимедийными элементами, чтобы не снизить результативность преподаваемого материала. При создании мультимедийной презентации необходимо решить задачу: как при максимальной информационной насыщенности продукта обеспечить максимальную простоту и прозрачность организации материала.

**Текст на слайде зрители практически не воспринимают.** Поэтому в презентациях лучше оставить текст только в виде имен, названий, числовых значений, коротких цитат. Текстовая информация заменяется схемами, диаграммами, рисунками, фотографиями, анимациями, фрагментами фильмов. Если невозможно избежать текстовой информации, то на слайде должно присутствовать не более трех мелких фактов и не более одного важного. Кроме того, понятия и абстрактные положения до сознания зрителя доходят легче, когда они подкрепляются конкретными фактами, примерами и образами; и потому для раскрытия их необходимо использовать различные виды наглядности. В то же время возможно только необходимое использование анимации и эффектов.

Лучше избегать обилия цифр. **Числовые величины** имеет смысл заменить сравнениями. Однако на этом пути тоже необходимо соблюдать чувство меры. Опыт работы показывает, что поток одних только ярких изображений воспринимается тоже не очень хорошо.

Хороший результат по переключению внимания дает **применение видеофрагментов**, особенно озвученных. Они почти всегда вызывают оживление в аудитории. Зрители устают от голоса одного лектора, а здесь внимание переключается, и тем самым поддерживается острота восприятия.

### 3. Разработка дизайна презентации

Важным моментом является выбор общего стиля презентации, унифицированной структуры и формы представления учебного материала на всем уроке. Стиль включает в себя:

- 1 - общую схему шаблона: способ размещения информационных блоков;
- 2 - общую цветовую схему дизайна слайда;
- 3 - цвет фона или фоновый рисунок, декоративный элемент небольшого размера и др.;

- 4 - параметры шрифтов (гарнитура, цвет, размер) и их оформления (эффекты),

- 5 - способы оформления иллюстраций, схем, диаграмм, таблиц и др.

Дизайн презентации должен **соответствовать ситуации**. Вся презентация должна выполняться **в одной цветовой палитре**, что создает ощущение связности, преемственности, стильности, комфортности. Для сохранения единообразия презентации начинающим пользователям лучше использовать шаблон презентации PowerPoint.

Рекомендуется выделять отдельные куски текста цветом; отдельные ячейки таблицы или всю таблицу цветом (фон ячейки или фон таблицы). Вся презентация выполняется в одной цветовой палитре, обычно на базе одного шаблона.

Очень важным является **фон слайдов**. Являясь элементом заднего (второго) плана, фон должен выделять, оттенять, подчеркивать информацию, находящуюся на слайде, но не заслонять ее. Серьезные презентации не должны быть пестрыми, содержать яркие, «ядовитые» цвета и менять цветовую гамму от слайда к слайду. Если презентация состоит из нескольких больших тем, то каждая тема может иметь свою цветовую гамму, но не сильно отличаться от общей цветовой гаммы презентации. Не стоит делать фон слишком пестрым, это отвлекает аудиторию и затрудняет чтение текста.

Для фона предпочтительны холодные тона или нейтральные тона: светло-розовый, серо-голубой, желто-зеленый, коричневый. Легкие пастельные тона лучше подходят для фона, чем белый цвет. С другой стороны, белое пространство признается одним из сильнейших средств выразительности.

Поскольку фон создает определенное настроение у аудитории и должен соответствовать теме презентации, то иногда целесообразно использование "тематического" фона: сочетание цветов, несущие смысловую нагрузку и т. п.

После ввода текста необходимо определиться с его расположением на каждом слайде, продумать его форматирование, т.е. определить размер, цвет шрифта, заголовков и основного текста. При подборе цвета текста помните, что **текст должен быть «читаем»**, т.е. фон слайдов не должен «глушить» текст.

Не рекомендуется использовать **переносы слов**, а также наклонное и вертикальное расположение подписей и текстовых блоков.

Учитывая, что **шрифты без засечек** – гладкие, плакатные – (типа **Arial, Tahoma, Verdana** и т.п.) легче читать с большого расстояния, чем шрифты с засечками (типа Times), то для основного текста предпочтительно использовать плакатные шрифты; для заголовка можно использовать декоративный шрифт, если он хорошо читаем и не контрастирует с основным шрифтом.

Рекомендуемые размеры шрифтов: для заголовков 32-50, оптимально – **36**; для основного текста: 18 – 32, оптимально – **24**.

Не следует злоупотреблять прописными буквами (они читаются хуже строчных), поэтому их допустимо использовать только для смыслового выделения небольших фрагментов текста.

Наиболее **важный материал**, требующий обязательного усвоения, желательно **выделить ярче** для включения ассоциативной зрительной памяти. Для выделения информации следует использовать цвет, жирный и/или курсивный шрифт. Выделение подчеркиванием обычно ассоциируется с гиперссылкой, поэтому использовать его для иных целей не рекомендуется.

Целесообразно применение различных маркеров (◆▶●■) для выделения элементов текста (**маркированные списки**).

### Подготовка медиафрагментов

Тексты презентации не должны быть большими. Выгоднее использовать сжатый, информационный стиль изложения материала. Нужно будет суметь вместить максимум информации в минимум слов, привлечь и удержать внимание обучаемых. Однако в мультимедийной презентации может содержаться дополнительный материал, а также материал для углубленного изучения темы. Профессионалы по разработке презентаций советуют использовать на слайде не более тридцати слов и пяти пунктов списка.

#### Рекомендуется:

- 1 - использование коротких слов и предложений, минимум предлогов, наречий, прилагательных;
- 2 - использование нумерованных и маркированных списков вместо сплошного текста;
- 3 - горизонтальное расположение текстовой информации, в т.ч. и в таблицах;
- 4 - каждому положению, идее должен быть отведен отдельный абзац текста;
- 5 - основную идею абзаца располагать в самом начале – в первой строке абзаца;
- 6 - идеально, если на слайде только заголовок, изображение (фотография, рисунок, диаграмма, схема, таблица и т.п.) и подпись к ней.

Размещенные в презентации графические объекты должны быть, в первую очередь, оптимизированными, четкими и с хорошим разрешением. Графические объекты не располагаются в середине текста, это плохо смотрится.

### Тестирование-проверка, доводка презентации

- 1) проверка на работоспособность всех элементов презентации;
- 2) проверка визуального восприятия презентации сторонними наблюдателями, в том числе с экрана.

Доводка презентации заключается в неоднократном просмотре своей презентации, определении временных интервалов, необходимых аудитории для просмотра каждого слайда, и времени их смены. Помните, что слайд должен быть на экране столько времени, чтобы аудитория могла рассмотреть, запомнить, осознать его содержимое. Между тем большой интервал между сменами слайдов снижает интерес. Возможно, при окончательном просмотре вам придется поменять местами некоторые слайды для создания более логической структуры презентации или внести в нее другие коррективы.

Форма ОТЧЕТНОСТИ: отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

Основная литература [1,2,5].

Контрольные вопросы для самопроверки

1. Как строить график в Power Point?
2. Как выполнять рисунки, текстовые фрагменты, оформлять титульную страницу?
3. Какими средствами можно пользоваться при трансляции созданной программы?

### **Лабораторная работа №10. QBASIC. Ввод и вывод данных Программирование алгоритмов линейной структуры**

Цель работы – освоить приёмы написания программ, научиться пользоваться командами.

Порядок выполнения:

1. Присвоить переменной **a** значение 0.3
2. Вывести на экран значение переменной **a**
3. Присвоить переменной **b** значение 5
4. Вывести на экран значение переменной **b**
5. Присвоить переменной **c** значение, равное сумме переменных **a** и **b**
6. Вывести на экран значение переменной **c**
7. Возвести значение переменной **a** в степень **b**, и присвоить полученной значению переменной **d**
8. Вывести на экран значение переменной **d**
9. Переменной **e** присвоить значение, равное синусу **a**
10. Вывести на экран значение переменной **e**
11. Нарисовать блок-схему программы

#### **Общие сведения**

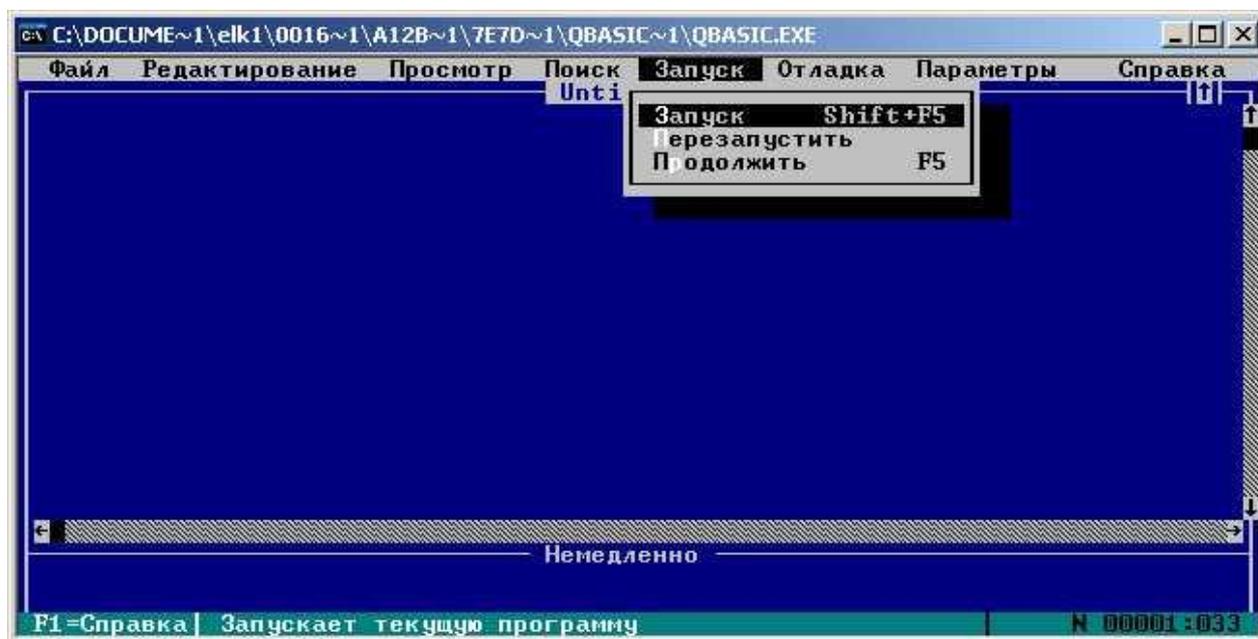
Лабораторные работы в разделе «Алгоритмизация и программирование» выполняются в среде программирования Microsoft Quick Basic 4.5. Язык Quick Basic (QBASIC – Beginner's All-purpose Instruction Code) разработан Джоном Кимини и Томасом Куртцем.

QBASIC занимает особое место среди всех языков высокого уровня. С самого начала он задумывался как универсальный язык для начинающих.

Вот некоторые достоинства QBASIC (с точки зрения массового пользователя):

- 1) простота синтаксиса;
- 2) простота организации данных и управляющих структур;
- 3) большое число встроенных команд и функций, позволяющих без труда выполнять такие операции, как управление текстовым и графическим экраном, обработка символьных строк и т.п.)

Особым достоинством QBASIC следует считать возможность работы в режиме интерпретации, который резко упрощает процесс отладки программ: исполнение почти каждой команды можно проверить сразу после написания (Shift + F5).



Программа на языке Бейсик записывается в виде последовательности символов, к числу которых относятся латинские и русские буквы, арабские цифры, знаки препинания ( , ; : " ' ? ) , знаки операций ( \* / - + < > <= >= = < > . ), специальные символы ( % & ! # \$ ).

Для обозначения исходных данных и результатов вычислений употребляются переменные. Последовательность латинских букв и цифр, начинающаяся с буквы, называется **идентификатором** или **именем переменной**.

Числа в программе записываются в десятичной системе, вместо запятой в десятичных дробях пишется точка: 0, - 17, 0.25, - 34.85.

**Переменные и числа** - это простейшие частные случаи выражения. Более сложные выражения строятся из чисел и переменных с помощью знаков сложения, вычитания, умножения, деления, возведения в степень ( + , - , \* , / , ^ ).

При вычислении значений выражений действуют обычные **правила старшинства операций**:

- 1) возведение в степень ^
- 2) умножение, деление \*, /
- 3) сложение, вычитание +, -

**Действия в арифметических выражениях выполняют слева направо** в зависимости от их приоритета. Для того чтобы изменить естественный порядок действий используются круглые скобки. Выражения в круглых скобках выполняются **в первую очередь**.

В выражении могут быть использованы следующие **встроенные функции**:

ABS (x) – модуль x –  $|x|$

SQR (x) – корень квадратный из x ( $\sqrt{x}$ )

INT (x) – целая часть x

SIN (x) – синус x (аргументом служит радианная мера угла)

COS (x) – косинус x

TAN (x) – тангенс x

ATN (x) – арктангенс x

LOG (x) – натуральный логарифм x

EXP (x) – экспонента x

SGN (x) – определение знака числа x

Полное описание всех функций и операторов можно найти во встроенной справочной системе QBASIC.



QBASIC – DOS-приложение. Запустить QBASIC можно несколькими способами. Самый простой – войти в каталог с приложением, выделить курсором файл qb.exe и нажать Enter. Однако, в этом случае придется настраивать QBASIC на работу с каталогом, в котором хранятся Ваши программы, так как хранить свои программы вместе с файлами QBASIC категорически не рекомендуется!

Второй способ – сделать активным нужный каталог и набрать в командной строке qb.exe (или просто qb) и нажать Enter. Чтобы этот вариант работал, нужно внести описание пути к файлу qb.exe в autoexec.bat.

После запуска программы qb.exe экран выглядит следующим образом



Основное поле называется окном редактора, в этом окне производится набор текста программ. В верхней строке экрана расположены слова, образующие *главное меню*. QBASIC может работать в режиме сокращенного и полного меню. Переход в режим полного меню – **Full Menus** находится в пункте **Options** главного меню.

В середине следующей строки расположено имя программы (точнее, имя файла с этой программой). Пока не задано имя, программа будет называться “Untitled”.

В пункте **File** главного меню нажмем клавишу Alt – при этом слово **File** станет светлым на черном фоне. Если теперь нажать Enter, то откроется выпадающее меню с новыми пунктами.

Сокращенное меню
New Program
Open Program
Save As...
Print...
Exit

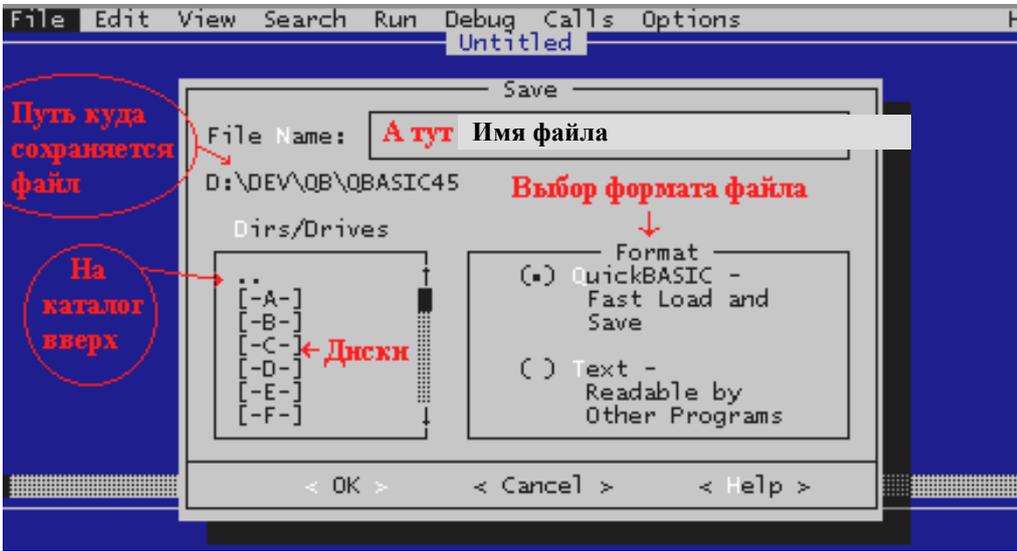
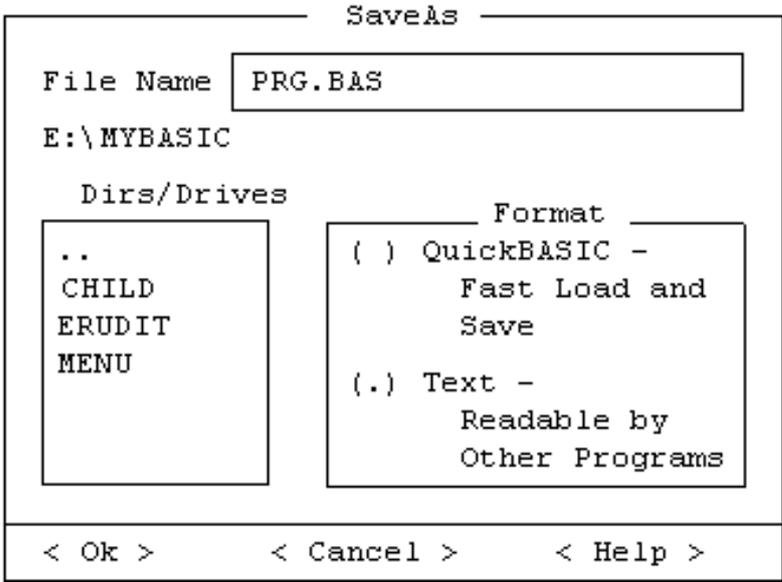
Полное меню
New Program
Open Program
Merge...
Save
Save As...
Save All
Create File...
Load File...

Unload File...
Print...
DOS Shell
Exit

- New Program** - перейти к набору новой программы.
- Open Program...** - загрузить готовую программу.
- Save As...** - записать программу на диск с тем же или новым именем.
- Print...** - напечатать текст программы на принтере.
- Exit** - выйти из среды программирования QBASIC.

Если при наборе строки операторы писать строчными буквами, то после нажатия на Enter все они (если в строке нет ошибок!) запишутся прописными буквами. Этот прием помогает заметить ошибку в имени оператора даже при отсутствии предупреждения.

Для сохранения программы на диске необходимо активировать главное меню (нажать и отпустить клавишу Alt), раскрыть пункт File и выбрать Save As... Многоточие обозначает, что при выборе такого пункта QBASIC вступает в диалог с пользователем. В данном случае на экране появится такое окно:



Не следует записывать свои программы в каталог QBASIC. Предположим, что они хранятся в каталоге C:\QBASIC. Если при запуске qb.exe активным был именно этот каталог, то все пункты меню, работающие с диском, будут использовать его по умолчанию.

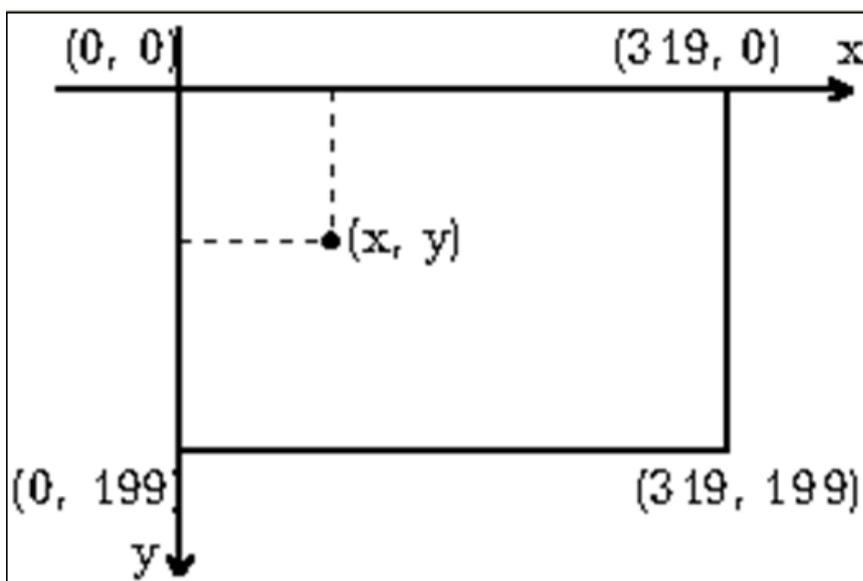
В области с именем File Name вводится имя программы. (Расширение .BAS можно не набирать, QBASIC добавляет его автоматически.) Прежде, чем нажать Enter, необходимо задать, в каком виде текст программы будет записан на диск. Если в области Format точка в круглых скобках стоит возле слова Text, то на диске окажется обычный текст, с которым могут работать и другие программы. Если же точка стоит возле слова QuickBASIC, то записанный на диск текст сможет прочитать и понять только сам QBASIC! Чтобы попасть в область Format, необходимо нажать несколько раз клавишу Tab.

Для выполнения набранной программы из главного меню необходимо раскрыть пункт Run, выбрать в выпавшем окне пункт Start (Shift+F5) и нажать клавишу ввода. Сочетание Shift+F5, написанное возле слова Start – это обозначение так называемой “горячей клавиши”. Нажатие такой комбинации клавиш выполняет те же действия, что и соответствующий пункт меню, но “напрямую” – непосредственно из окна редактора программы, без входа в главное меню.

### Работа с монитором в текстовом режиме

Монитор может работать в двух режимах: текстовом и графическом. В языке программирования QBASIC предусмотрены способы вывода информации на экран в обоих режимах. По умолчанию QBASIC работает с монитором в текстовом режиме. В этом режиме на экран можно вывести все символы, которые есть на клавиатуре, и символы псевдографики. В текстовом режиме экран имеет 25 строк, в каждой строке 80 позиций (однако, с помощью оператора WIDTH 40 можно получить 40 “широких” позиций) для вывода символов, следовательно, максимальное количество символов, которое можно разместить на экране в текстовом режиме, равно  $2000=25 \times 80$

Оператор SCREEN *номер режима* переводит режим работы экрана из текстового в графический с заданным номером. Если оператор SCREEN не указан, то подразумевается текстовый режим. Текстовый режим имеет номер 0, поэтому, если нужно из графического режима перейти в текстовый, следует использовать SCREEN 0. Иногда при возврате из графического режима (например, из SCREEN 7) символы текста становятся широкими. Вернуть им нормальные размеры можно только оператором WIDTH 80. Графические режимы различаются разрешением (количеством “точек”) экрана и количеством возможных цветов. Например, SCREEN 7 устанавливает разрешение экрана в 320 точек по горизонтали и 200 точек по вертикали.



Каждый редактор текста обязан уметь работать не только с отдельными символами, но и с более крупными фрагментами текста. Фрагмент может занимать часть одной строки или охватывать одну или несколько строк. Для выделения части строки следует поставить курсор на начало фрагмента и несколько раз нажать комбинацию клавиш Shift+[→]. Можно выделить текст влево от курсора, если несколько раз нажать на Shift+[←]. Аналогично выделяются и строки, только нужно воспользоваться комбинацией клавиш Shift+[↑] или Shift+[↓].

Работа с фрагментами сосредоточена в пункте **Edit** главного меню:

**Cut** (Shift+Del) – удалить выделенный фрагмент. Удаленный фрагмент исчезает с экрана, но временно сохраняется в специальной области памяти. В буфере хранится только последний удаленный фрагмент.

**Copy** (Ctrl+Ins) – выделенный фрагмент копируется в буфер, удаления с экрана не происходит.

**Paste** (Shift+Ins) – содержимое буфера вставляется в текст программы. Место вставки указывает курсор.

**Clear** (Del) – выделенный фрагмент удаляется без сохранения в буфере.

С помощью пунктов Cut и Paste можно перенести любой фрагмент в другое место программы; Copy и Paste позволяют “размножить” один и тот же фрагмент. Если нужный фрагмент занимает ровно одну строку, то ее можно не выделять. Строка удаляется комбинацией Ctrl+Y, но при этом, в отличие от других текстовых редакторов, попадает в буфер и может быть восстановлена в том же или другом месте!

### Ввод и вывод данных

Оператор присваивания имеет общий вид:

$$\text{имя переменной} = \left\{ \begin{array}{l} \text{èìÿ äďóãíé ïäðàìáííé} \\ \text{àďèòìàòè÷á ñêîâ âûðàæáíéâ} \\ \text{÷èñëí} \end{array} \right.$$

**Переменная** – любая переменная, имя которой может быть длиной до 40 символов и должно начинаться с буквы. Допустимые символы A–Z, 0–9 и точка (.).

**Арифметическое выражение** – любое выражение, которое имеет присваиваемое значение.

В результате выполнения оператора присваивания переменной, стоящей слева от знака равенства присваивается значение другой переменной или значение арифметического выражения или численное значение. Для того чтобы оператор присваивания мог быть выполнен, необходимо, чтобы все переменные в выражении имели некоторые значения и были заданы выше.

### ПРИМЕР

1. A = 0
2. C = 2  
   B = C
3. A = 2: B=3  
   C = (A + B) ^ 2 / 5

**Оператор PRINT** – выводит данные на экран или записывает их в файл.

**Оператор LPRINT** выводит данные на печать (на принтер).

PRINT [#номер\_файла%,] [выражения] [{; | ,}]

LPRINT [выражения] [{; | ,}]

– **номер\_файла%** – номер открытого файла. Если номер файла не указан, PRINT выводит данные на экран.

– **выражения** – одно или несколько числовых или символьных выражений для печати.

– {; | ,} – определяет, где начинается следующий вывод:

; – печать сразу после последнего значения

, – печать в начале следующей зоны печати (ширина зоны печати 14 символов).

Если при работе в среде QBasic 4.5 вместо слова PRINT ставить знак вопроса "?", а затем нажать Enter, то слово PRINT само появится на экране.

Для вставки комментария в программу используется оператор **REM** или апостроф (').

REM комментарий  
' комментарий

Комментарий – любой текст. При работе программы комментарии игнорируются, если только они не содержат метакоманды. Комментарий может быть вставлен в строке после выполняемого оператора, если перед комментарием стоит символ апострофа (') или перед REM стоит двоеточие (:).

#### ПРИМЕР

```
REM Это комментарий.  
' Это тоже комментарий.  
PRINT "Тест1" 'Это комментарий после оператора PRINT.
```

#### Программирование алгоритмов линейной структуры

Программа имеет линейную структуру, если все операторы (команды) выполняются последовательно друг за другом.

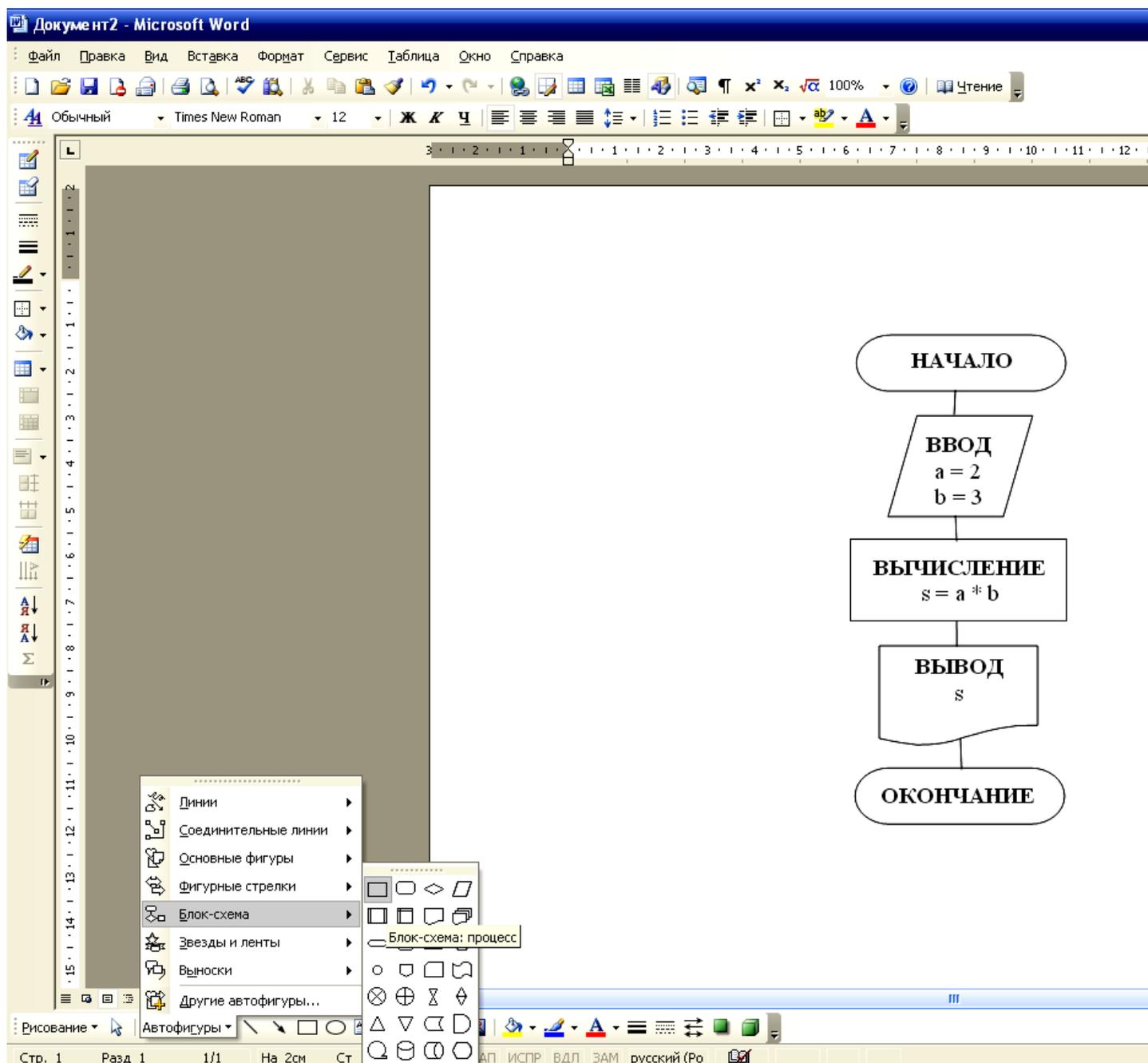
#### ПРИМЕР

Вычислите площадь прямоугольника по его сторонам.

```
REM Площадь прямоугольника  
a = 2  
b = 3  
s = a * b  
PRINT "Площадь равна: ", s  
END
```

В графическом способе изображения структур алгоритмов используется набор блочных символов, соединённых линиями передачи управления (перехода). Этот способ иногда ещё называется способом блок-схем. Основными его особенностями являются краткость и

выразительность. При составлении блок-схем удобно использовать текстовый редактор Microsoft Word. На панели «Рисование» имеется инструмент «Автофигуры». Используем инструмент «Блок-схема».



**Форма ОТЧЕТНОСТИ:** отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

**Основная литература** [1,2,6].

**Контрольные вопросы для самопроверки**

1. Какие команды знаете?
2. Как выполнять рисунки, текстовые пояснения?
3. Какими средствами можно пользоваться при отладке созданной программы?

**Лабораторная работа №11. QBASIC. Программирование алгоритмов разветвлённой сети.**

**Цель работы** – освоить приёмы написания программ, научиться пользоваться командами.

## Порядок выполнения

Любой язык программирования опирается на трех китов: следование, ветвление, повторение (цикл).

### **Ветвление**

Ветвление в большинстве языков программирования, в том числе и в Бейсике, реализуется с помощью условного оператора, в котором используются ключевые слова **IF**, **THEN**, **ELSE**, в переводе на русский они означают – ЕСЛИ, ТОГДА, ИНАЧЕ.

В программе на Бейсике оператор ветвления может записываться в одну строку

IF *условие* THEN *ветвь "то"* (неполное ветвление)

или

IF *условие* THEN *ветвь "то"* ELSE *ветвь "иначе"* (полное ветвление).

И в несколько строк

IF *условие* THEN  
*ветвь "то"*  
END IF (неполное ветвление)

или

IF *условие* THEN  
*ветвь "то"*  
ELSE *ветвь "иначе"*  
END IF (полное ветвление)

Ветвь “то” исполняется, если при проверке окажется, что условие верно. Если же окажется, что условие неверно, то будет исполнена ветвь “иначе”.

### **Ввод с клавиатуры**

Оператор **INPUT** *список\_переменных* принимает числовые или текстовые значения с клавиатуры и присваивает их переменным из списка. Программа, дойдя до оператора INPUT, выводит на дисплей знак вопроса и ждет ввода с клавиатуры. Для общения с клавиатурой используется небольшая область памяти (буфер клавиатуры). При нажатии клавиши соответствующий символ (точнее, его код) попадает в буфер клавиатуры, оператор INPUT забирает его оттуда и отображает на экране. Ввод заканчивается при нажатии клавиши Enter.

Имена переменных, предназначенных для хранения текстовых значений, должны заканчиваться знаком \$. Имена переменных, хранящих числовые значения, могут в зависимости от типа значения заканчиваться знаком !, # или &. При отсутствии этих знаков в имени переменной считается, что она имеет тип, соответствующий восклицательному знаку.

Пример:

В программе                    INPUT x, y  
                                  PRINT x, y

На дисплее                    ? 30, 47 ↵  
                                  30 47

В результате переменная “x” получит значение 30, а переменная “y” – значение 47.

Если повторно запустить программу и ввести только одно значение, например, 30, то Quick Basic предложит повторить ввод – выдаст сообщение “Redo from start”. То же самое сообщение мы получим, если захотим ввести значений больше, чем переменных в списке оператора INPUT.

**Базовая структура ветвление** обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

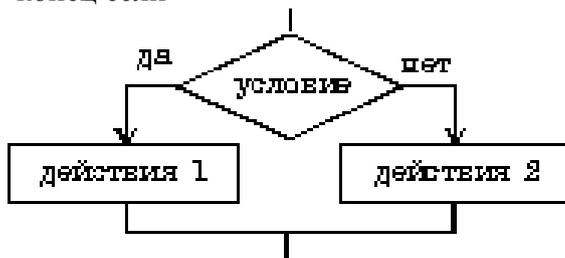
Структура **ветвление** существует в четырех основных вариантах:

- *если-то*;
- *если-то-иначе*;
- *выбор*;
- *выбор-иначе*.

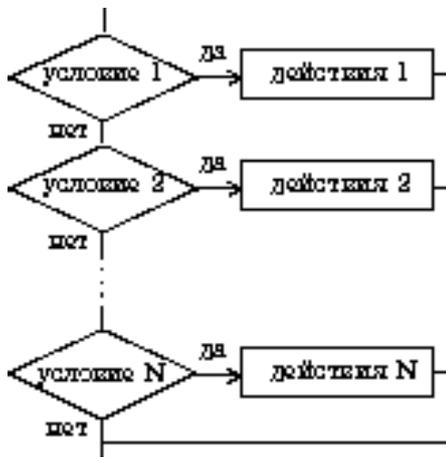
1) *если-то*  
если условие  
то действия  
конец если



2) *если-то-иначе*  
если условие  
то действия 1  
иначе действия 2  
конец если



3) *выбор*  
выбор  
при условии 1: действия 1  
при условии 2: действия 2  
.....  
при условии N: действия N  
конец выбора



4) *выбор-иначе*

выбор

при условии 1: действия 1

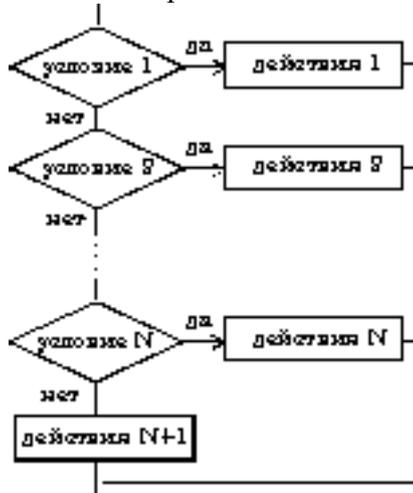
при условии 2: действия 2

.....

при условии N: действия N

иначе действия N+1

конец выбора



**Задание на лабораторную работу**

Выполните заданные по вариантам действия.

Текст программы выполнять в среде программирования Microsoft Quick Basic 4.5.

Отрисовку блок-схем выполнять текстовом редакторе Microsoft Word.

Вариант №1	Вариант №2
<p>a=15, b= -0.5, c=13.7 вести с клавиатуры число x</p>	<p>a=-0.7, b= -0.2, c=3.8 вести с клавиатуры число x</p>

$y = \begin{cases} ax^3 + bx + c, & \text{если } x < 0 \\ \frac{a}{c} + b, & \text{если } x = 0 \\ -\frac{b}{a}x + c, & \text{если } x > 0 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>	$y = \begin{cases} \frac{a\sqrt[3]{x}}{c} + bx^2, & \text{если } x < 3 \\ \left(\frac{a}{c}\right)^3 + b, & \text{если } x = 3 \\ b \cdot \operatorname{tg}(x) - c^a, & \text{если } x > 3 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>
<p><b>Вариант №3</b></p> <p><math>a=9, b=5, c=3</math> ввести с клавиатуры число <math>x</math></p> $y = \begin{cases} a \sin(x^3) + b \cos(x^2), & \text{если } x < -2 \\ c, & \text{если } x = -2 \\ a \sin(x^3) + b \cos(x^2), & \text{если } x > -2 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>	<p><b>Вариант №4</b></p> <p><math>k=9, m=5, p=3</math> ввести с клавиатуры число <math>x</math></p> $y = \begin{cases} k \cdot \operatorname{tg}(x^m) + p, & \text{если } x < 5 \\ kmp, & \text{если } x = 5 \\ m \log(x) + p, & \text{если } x > 5 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>
<p><b>Вариант №5</b></p> <p><math>k=1, m=2, p=3</math> ввести с клавиатуры число <math>x</math></p> $y = \begin{cases} -k x  + p, & \text{если } x < 0 \\ m, & \text{если } x = 0 \\ p\sqrt{x}, & \text{если } x > 0 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>	<p><b>Вариант №6</b></p> <p><math>k=-3, m=4, p=7</math> ввести с клавиатуры число <math>x</math></p> $y = \begin{cases} e^x + kx, & \text{если } x < 0 \\ \frac{k}{p}, & \text{если } x = 0 \\ \sin(x) - pm, & \text{если } x > 0 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>
<p><b>Вариант №7</b></p> <p><math>a=-3, b=4, c=7</math> ввести с клавиатуры число <math>x</math></p> $y = \begin{cases} a\sqrt{-x} + b\sqrt[3]{x} + c, & \text{если } x < 0 \\ c, & \text{если } x = 0 \\ ax^2 + bx^3 + c, & \text{если } x > 0 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>	<p><b>Вариант №8</b></p> <p><math>a=2, b=0.4, c=7.6</math> ввести с клавиатуры число <math>x</math></p> $y = \begin{cases} a \operatorname{tg}(x) + c, & \text{если } x < 3 \\ \frac{b}{c}, & \text{если } x = 3 \\ (a+b)x + c, & \text{если } x > 3 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>
<p><b>Вариант №9</b></p> <p><math>a=1, b=8, c=-13</math> ввести с клавиатуры число <math>x</math></p>	<p><b>Вариант №10</b></p> <p><math>a=0.5, b=3, c=2</math> ввести с клавиатуры число <math>x</math></p>

$y = \begin{cases} ax^2 + c, & \text{если } x < -5 \\ b, & \text{если } x = -5 \\ cx - b, & \text{если } x > 5 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>	$y = \begin{cases} \frac{ax + b}{\sqrt{x}c}, & \text{если } x < 1.5 \\ \frac{b + c}{a}, & \text{если } x = 1.5 \\ \frac{(a - x)c}{b}, & \text{если } x > 1.5 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>
<p><b>Вариант №11</b></p> <p><math>a=-2, b=1, c=6</math> ввести с клавиатуры число <math>x</math></p> $y = \begin{cases} \left(\frac{x}{a-b}\right)^c, & \text{если } x < 0 \\ \frac{ab}{c}, & \text{если } x = 0 \\ \sqrt{x}c, & \text{если } x > 0 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>	<p><b>Вариант №12</b></p> <p><math>d=0.7, e=2, f=-3</math> ввести с клавиатуры число <math>x</math></p> $y = \begin{cases} \frac{d + f}{x}, & \text{если } x < 0 \\ e + \cos f, & \text{если } x = 0 \\ \left(\left \frac{d \cos x}{e \sin x}\right \right)^x, & \text{если } x > 0 \end{cases}$ <p>вывести на экран значение <math>y</math> нарисовать блок-схему программы</p>

Форма ОТЧЕТНОСТИ: отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

Основная литература [1,2,6].

Контрольные вопросы для самопроверки

1. Что такое разветвление структуры алгоритма?
2. Какие условия задают направление вычислений при ветвлении программы?

### Лабораторная работа №12. QBASIC. Программирование циклических алгоритмов.

Цель работы – освоить навыки работы с клавиатурой, ознакомиться с основными элементами интерфейса Microsoft Word научиться пользоваться основными средствами оформления документов.

Порядок выполнения:

- 1) Очистить экран
- 2) Присвоить переменной  $a$  значение 3
- 3) Присвоить переменной  $b$  значение 10
- 4) Присвоить переменной  $c$  значение -7
- 5) Используя цикл с предусловием, подсчитать значения переменной при изменении переменной от начального значения  $x_n = 0$  до конечного значения переменной  $x_k = 15$  через шаг  $dx = 1$ , с условием, что

$$y = \begin{cases} a, & \text{при } x < 7 \\ b, & \text{при } x = 7 \\ c, & \text{при } x > 7 \end{cases}$$

- 6) Вывести на экран значение переменной  $y$  при соответствующем значении переменной  $x$ .

7) Нарисовать блок-схему программы

### Операторы цикла в Бейсике

Операторов цикла в Бейсике несколько. Есть два оператора цикла типа «повторяй, пока выполняется условие» (операторы **DO WHILE LOOP**):

1) проверка условия в начале цикла

**DO WHILE** *условие*

*тело\_цикла*

**LOOP**

2) проверка условия в конце цикла

**DO**

*тело\_цикла*

**LOOP WHILE** *условие*

Из цикла **DO** можно выйти досрочно по команде **EXIT DO**, даже если условие выполнения продолжает оставаться истинным.

### Пример цикла с предусловием

<p>1) Очистить экран 2) Присвоить переменной <b>a</b> значение 1 3) Присвоить переменной <b>b</b> значение 5 4) Присвоить переменной <b>c</b> значение -3 5) Используя цикл с предусловием, подсчитать значения переменной при изменении переменной от начального значения <math>x_n = 1</math> до конечного значения переменной <math>x_k = 10</math> через шаг <math>dx = 1</math>, с условием, что</p> $y = \begin{cases} \mathbf{a}, & \text{при } x < 5 \\ \mathbf{b}, & \text{при } x = 5 \\ \mathbf{c}, & \text{при } x > 5 \end{cases}$ <p>6) Вывести на экран значение переменной <b>y</b> при соответствующем значении переменной <b>x</b>.</p>	<pre>CLS a = 1 b = 5 c = -3 x0 = 1 x1 = 10 dx = 1 x = x0 DO WHILE x &lt;= x1 IF x &lt; 5 THEN y = a IF x = 5 THEN y = b IF x &gt; 5 THEN y = c PRINT "при x="; x; " y="; y x = x + dx LOOP</pre>
---	--

### Пример цикла с постусловием

<p>1) Очистить экран 2) Присвоить переменной <b>a</b> значение 1 3) Присвоить переменной <b>b</b> значение 5 4) Присвоить переменной <b>c</b> значение -3 5) Используя цикл с постусловием, подсчитать значения переменной при изменении переменной от начального значения <math>x_n = 1</math> до конечного значения переменной <math>x_k = 10</math> через шаг <math>dx = 1</math>, с условием, что</p>	<pre>CLS a = 1 b = 5 c = -3 x0 = 1 x1 = 10 dx = 1 x = x0 DO IF x &lt; 5 THEN y = a IF x = 5 THEN y = b</pre>
---	--

$y = \begin{cases} a, & \text{при } x < 5 \\ b, & \text{при } x = 5 \\ c, & \text{при } x > 5 \end{cases}$ <p>6) Вывести на экран значение переменной <math>y</math> при соответствующем значении переменной <math>x</math>.</p>	<pre>IF x &gt; 5 THEN y = c PRINT "при x="; x; " y="; y x = x + dx LOOP while x &lt;= x1</pre>
--	--

### Цикл с параметром (цикл с перебором):

FOR параметр = начальное\_значение TO конечное\_значение [STEP шаг]

*тело\_цикла*

NEXT параметр

Заметим, что часть заголовка цикла - *STEP шаг* - может отсутствовать (поэтому она взята в квадратные скобки!). В этом случае считается, что параметр цикла увеличивается каждый раз на 1. Параметр - это имя некоторой переменной величины. Для каждой переменной Quick Basic отводит место в оперативной памяти, где и хранится ее текущее значение.

Оператор цикла работает следующим образом: при входе в цикл параметру присваивается начальное значение. Проверяется условие “значение параметра не больше конечного значения” при положительном шаге изменения параметра или условие “значение параметра не меньше конечного значения” при отрицательном шаге. Если проверяемое условие истинно, то выполняется тело цикла, затем к значению параметра прибавляется шаг изменения и происходит возврат к проверке условия. При невыполнении условия управление передается оператору, следующему за признаком конца цикла FOR – NEXT. Из этого цикла также можно выйти, не дожидаясь конца повторений с помощью оператора EXIT FOR.

### Рисование точек и линий

Оператор **PSET** ( $x, y$ )[, *цвет*] закрашивает точку экрана заданным цветом.

Оператор **LINE** ( $x1, y1$ )-( $x2, y2$ )[, *цвет*] рисует отрезок, соединяющий на экране точки с координатами ( $x1, y1$ ) и ( $x2, y2$ )

Оператор **LINE** - ( $x2, y2$ )[, *цвет*] рисует отрезок, соединяющий последнюю выведенную на экран точку с точкой ( $x2, y2$ ).

Цвет в этих операторах задается числом от 0 до 15. Если цвет в графических операторах не указан, то все линии рисуются белым цветом.

### Случайные числа

Случайное число - это число, величина которого не может быть заранее известна. Классический пример - число очков, выпадающих при бросании игрального кубика.

В Quick Basic имеется функция **RND**, которая возвращает случайное значение из интервала (0, 1).

Чтобы получать разные последовательности нужно использовать команду **RANDOMIZE TIMER**.

**RANDOMIZE** получает от функции **TIMER** значение системного времени и формирует значение для **RND**. При следующем запуске программы функция **TIMER** вернет другое значение, и получится другая последовательность чисел.

Для получения случайного целого числа  $x$  в пределах от 1 до  $N$  умножим неравенство  $0 < \text{RND} < 1$  на  $N > 0$ . При умножении на положительное число знак неравенства сохраняется, поэтому  $0 < N * \text{RND} < N$ . Чтобы получить целую часть результата воспользуемся функцией **INT**. Но **INT**( $N * \text{RND}$ ) изменяется в пределах от 0 до  $N-1$ . Осталось только прибавить единицу, чтобы получить окончательную формулу:

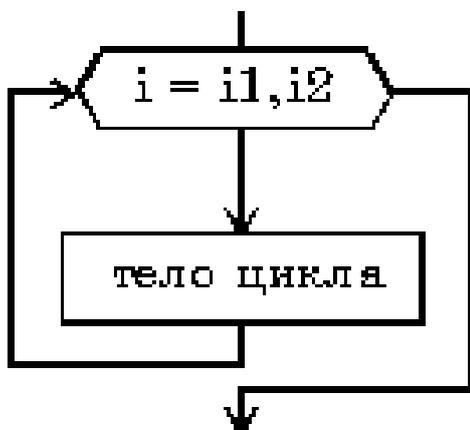
$$x = \text{INT}(N * \text{RND}) + 1$$

## Базовая структура цикл

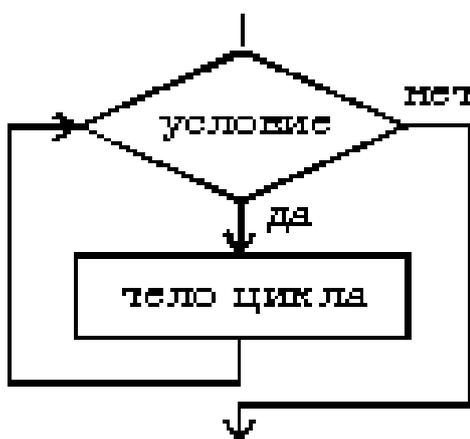
Базовая структура цикл. Обеспечивает многократное выполнение некоторой совокупности действий, которая называется телом цикла.

Алгоритм структуры цикл существует в трех основных вариантах:

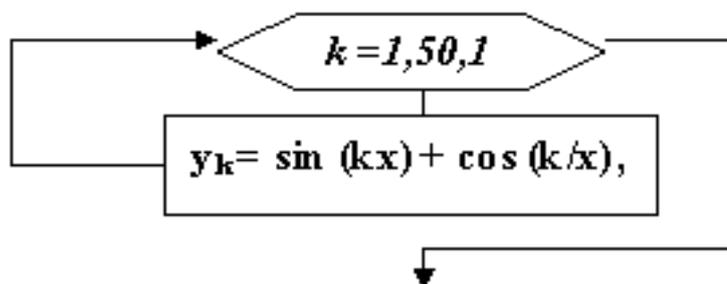
### 1) Цикл с перебором



### 2) Цикл с предусловием



### 3) Цикл с постусловием



Форма отчетности: отчет по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчет по лабораторному практикуму.

Основная литература [1,2,6].

Контрольные вопросы для самопроверки

1. Как строить циклические вычисления в программе?
2. Как выполнять начало цикла, его тело и перадресацию при окончании цикла?
3. Какими средствами можно пользоваться при окончании цикла?

## Лабораторная работа №13 QBASIC. Работа с массивами.

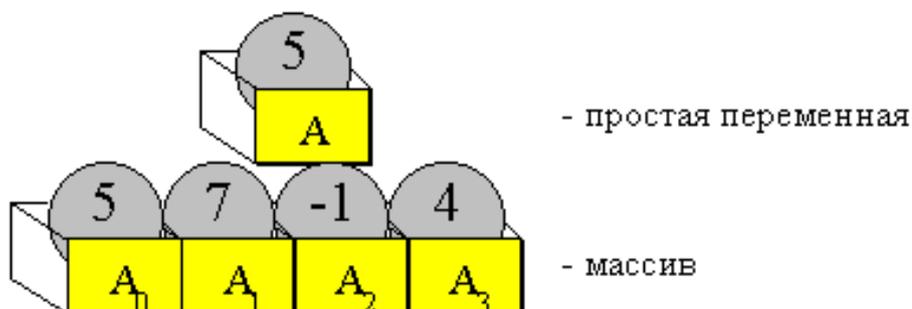
Цель работы – освоить приёмы написания программ-массивов, научиться пользоваться необходимыми командами.

### Порядок выполнения:

1. Сформировать массив **A** состоящий из 20 элементов и заполнить его случайными целыми числами в диапазоне [-10, 10]
2. Полученный массив распечатать
3. Найти сумму элементов массива
4. В массиве найти максимальный элемент и его порядковый номер
5. Отсортировать массив по возрастанию
6. Распечатать полученный массив

### **Понятие массива**

Проиллюстрируем наглядно понятия простой переменной и массива. Простую переменную можно представить в виде ящичка, на котором написано имя переменной (в нашем случае - **A**). В ящичек можно положить любое число - значение переменной (в нашем случае - 5).



Массив можно представить как ящик с перегородками. В каждый из получившихся “отсеков” можно положить свое число. Отсеки пронумерованы, поэтому имя каждого отсека состоит из двух частей: общего имени ящика и номера отсека. Но такие обозначения как  $A_0$ ,  $A_1$ ,  $A_2$  используются в математике, в языках программирования элемент массива обозначается  $A(0)$ ,  $A(1)$ ,  $A(2)$  и т.д.

Каждый массив должен быть описан в операторе **DIM** (от английского dimensions – размеры):

**DIM** описывает массив или указывает тип данных для переменных не массива.

**REDIM** описывает или изменяет размер динамического массива, стирая все предыдущие значения.

**DIM** [SHARED] переменная[(индекс)] [AS тип]

[,переменная[(индекс)] [AS тип]]...

**REDIM** [SHARED] переменная(индекс) [AS тип]

[,переменная(индекс) [AS тип]]...

■ **SHARED** Указывает, что переменные используются совместно всеми процедурами **SUB** или **FUNCTION** в модуле.

■ **переменная** Имя переменной массива.

■ **индекс** Размеры массива в следующем виде:

[низ TO] верх [, [низ TO] верх]...

низ Нижняя граница индекса массива. По умолчанию нижняя граница равна нулю.

верх Верхняя граница.

- AS тип      Описывает тип данных массива или переменной (INTEGER, LONG, SINGLE, DOUBLE, STRING или тип данных, описанный пользователем).
- DIM описывает либо статические, либо динамические массивы. Если область хранения массива не была описана в \$STATIC, \$DYNAMIC или COMMON, массивы, размер которых был задан в числах, являются статическими, а массивы, размер которых был задан в переменных - динамическими. REDIM всегда описывает динамические массивы.
- Область хранения статического массива выделяется при запуске программы и остается постоянной. Область хранения динамического массива выделяется при работе программы.

DIM имя\_массива(номер\_последнего\_элемента)

Поскольку нумерация элементов массива в Quick Basic начинается с нуля, то количество элементов массива (длина массива) оказывается на единицу больше номера последнего элемента. Например, DIM A(7) описывает массив из восьми элементов. Иногда, для удобства работы, мы будем “забывать” о нулевом элементе и указывать в описании массива его длину.

Элементы массива должны иметь значения одного и того же типа. Для указания типа элементов к имени массива добавляется знак \$, если элементы массива суть строки символов, или знак %, если все элементы массива - целые числа. (Подробнее о типах данных поговорим в соответствующем разделе.)

Указатели типа являются частью имени как обычной переменной так и массива, поэтому A% и A\$ - это имена *разных* переменных!

### Операторы GET и PUT

Оператор GET позволяет запомнить часть графического экрана в некотором массиве, а оператор PUT - вывести запомненное изображение на экран. Синтаксис оператора GET:

GET (x1, y1)-(x2, y2), массив - запомнить прямоугольную область.

Точки (x1, y1) и (x2, y2) - две вершины прямоугольника, лежащие на одной из его диагоналей. Размеры прямоугольника и его содержимое запоминаются в массиве.

Синтаксис оператора PUT:

PUT (x, y), массив [, способ\_вывода] - вывести на экран картинку из массива заданным способом.

В операторах GET и PUT будем использовать только массивы *целых* чисел. Размер массива должен быть достаточно большим, чтобы в него поместилась необходимая информация (сохранилась нужная картинка с экрана). В литературе приводятся точные формулы, но можно пользоваться более простой приближенной формулой

$$\text{размер\_массива} = 0.3 * dx * dy,$$

где  $dx = x2 - x1 + 1$ ,  $dy = y2 - y1 + 1$  – размеры (в пикселях) прямоугольной области, полностью закрывающую интересующую нас картинку, по горизонтали и вертикали.

Если размер массива, вычисленный по приближенной формуле, окажется недостаточным, Quick Basic выдаст предупреждение “Illegal function call” (“Неверный вызов функции”) и остановит выполнение программы на строке с оператором GET. В этом случае следует несколько увеличить размер массива и запустить программу еще раз.

Форма отчетности: отчет по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчет по лабораторному практикуму.

Основная литература [1,2,6].

### Контрольные вопросы для самопроверки

1. Что такое массив?
2. Что такое получение статистических данных о массиве?

### **Лабораторная работа №14. Работа с символьными переменными.**

Цель работы – научиться использовать строковые, т.е. символьные переменные, константы и выражения для обработки текстовой информации.

#### Порядок выполнения

1. подсчитать сколько раз среди символов заданной строки встречается буква, заданная с клавиатуры.

```
INPUT"ВВЕДИТЕ СТРОКУ";S$
INPUT"ВВЕДИТЕ ИСКОМУЮ БУКВУ";C$
K=0
FOR I=1 TO LEN(S$)
IF MID$(S$,I,1)=C$ THEN K=K+1
NEXT I
PRINT "БУКВА";C$;"ВСТРЕТИЛАСЬ";K;"РАЗ"
```

**Задача:** составить программу, которая печатает заданное слово задом наперед

```
INPUT"ВВЕДИТЕ СЛОВО"; S$
FOR I=LEN(S$) TO 1 STEP -1
? MID$(S$,I,1)
NEXT I
```

### **Символьные переменные**

Программы на языке Бейсик обрабатывают не только числовую, но и текстовую информацию (т.е. строки символов). Для этого используются строковые, т.е. символьные переменные, константы и выражения. В памяти хранятся в виде кодов.

Символьная константа-строка символов, заключенная в кавычки. Значением константы является последовательность, состоящая из символов, не считая кавычек. Внутри кавычек допускаются любые символы, не считая кавычек. Два идущих подряд знака кавычек задают пустую символьную строку.

У символьной переменной есть имя, которое оканчивается знаком \$. И значение в виде строки символов, которое может изменяться по ходу выполнения программы. Фактически имя символьной переменной в каждый момент времени указывает на некоторую область памяти, в которой находятся коды символов, составляющих текущее значение переменной.

Длина символьной переменной или константы 255 знаков (зависит от версии Бейсика).

*Символьный массив* – массив, элементами которого являются символьные строки.

Массив можно рассматривать как символьную переменную, снабженную одним или 2 индексами.

*Символьная функция* – функция, значениями которой являются строки символов или имя символьной функции оканч знаком \$.

*Символьное выражение* – символьная константа или переменная, ссылка на элемент символьного массива или значение символьной функции, а также результат любого их соединения при помощи знака + или & (операция конкатенации или сцепления строк).

### **Операции сравнения**

Строки считаются равными, если их длины равны и коды всех символов попарно совпадают. Если одна из строк совпадает с началом другой, но короче, решает код первого несовпадающего символа – меньше та строка, у которой он меньше.

Значения символьной переменной могут задаваться при помощи:

1. Операции присваивания.
2. INPUT
3. DATA – READ

```
A$="ЛРАБ"  
READ B$  
DATA "ИНФОРМАТИКА"  
C$=A$+LEFT$(B$,10)+"И"  
PRINT C$
```

Каков будет ответ, посмотреть на компьютере.

### **Основные стандартные функции**

Строка – произвольное символьное выражение.

Число – числовое.

ASC(строка) – выдает код первого символа строки.

```
PRINT"КОД БУКВЫ F=";ASC("F")  
CHR$(ЧИСЛО) - обратная функции ASC  
?"БУКВА С КОДОМ 70 - ЭТО":CHR$(70)
```

STR\$(число) – преобразует число в символьную строку, которая представляет собой запись числа в виде последовательности десятичных цифр (возможно со знаком и точкой)

```
S$="7*8="+STR$(7*8)
```

LEN(строка) – выдает длину значения символьного выражения

```
S$="7*8="+STR$(7*8)
```

```
?LEN(S$)
```

Учитываются все пробелы.

LEFT\$(строка, число) – выдает в качестве своего значения начальный отрезок заданной строки, те заданный вторым аргументом функции количество символов слева. Вырезка слева.

RIGHT\$(строка, число) – выдает в качестве своего значения конечный отрезок заданной строки, те заданный вторым аргументом функции количество символов справа. Вырезка справа.

```
? LEFT$("ИНФОРМАЦИЯ",7)+ RIGHT$("МАТЕМАТИКА",4)
```

MID\$(строка, число1, число2) – функция. Обобщает 2 предыдущие и позволяет получить любую подстроку заданной строки.

Число1 – задает начальную позицию подстроки в заданной строке,  
число2 – длину подстроки. Если заданная позиция находится за пределами строки или заданная длина неположительна, выдается пустая строка.

Если третий аргумент третий аргумент опущен или его значение превосходит количество символов от заданной позиции до конца строки, то выдает все эти символы.

```
X$="ПОБЕДА"
FOR I=2 TO 5
?MID$(X$,I,4)
NEXT I
```

Двойник (оператор)

MID\$(переменная\$, число1, число2)=строка,

где переменная обозначает произвольную символьную переменную.

Оператор присваивает указанной в нем строковой переменной значение = значению новой функции, здесь не указанной.

```
F$="4+??=29"
MID$(F$,3,2)="25"
?F$
```

**Задание:** подсчитать сколько раз среди символов заданной строки встречается буква, заданная с клавиатуры.

```
INPUT"ВВЕДИТЕ СТРОКУ";S$
INPUT"ВВЕДИТЕ ИСКОМУЮ БУКВУ";C$
K=0
FOR I=1 TO LEN(S$)
IF MID$(S$,I,1)=C$ THEN K=K+1
NEXT I
PRINT "БУКВА";C$;"ВСТРЕТИЛАСЬ";K;"РАЗ"
```

**Задача:** составить программу, которая печатает заданное слово задом наперед

```
INPUT"ВВЕДИТЕ СЛОВО";S$
FOR I=LEN(S$) TO 1 STEP -1
? MID$(S$,I,1)
NEXT I
```

Форма отчетности: отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

Основная литература [1,2,7].

Контрольные вопросы для самопроверки

1. Что такое символьная переменная?
2. Что такое стандартная функция?

**Лабораторная работа №15.** QBASIC. Работа с файлами данных.

Цель работы – освоить навыки работы с совокупностями числовых данных.

Порядок выполнения

### 1. Основные функции

Большинство программ работает с данными, которые можно вводить с клавиатуры и выводить на экран, или с данными, хранящимися в каких-либо файлах на дискетах или жестких дисках. Рассмотрим некоторые основные понятия. Прежде всего, следует выяснить, что собственно понимают под словом "файл". Это понятие можно определить так: **файл** – это область на носителе информации, где хранятся некоторые данные. Правда, это определение слишком обобщенное.

Нас будут интересовать лишь такие файлы, хранящиеся на носителях, которые можно обрабатывать программами QBASIC. Сравним файл с картотекой. В картотечном ящике, как правило, находится некоторое количество карточек, в противном случае – картотека пуста. Если к картотеке применить понятие файла, то может возникнуть ситуация, когда файл

окажется пустым. Однако файл существует, в нем только нет информации. Отдельные карточки картотеки называются записями. Так же, как карточки в ящике картотеки, записи в файле расположены последовательно: есть первая запись файла и последняя. На отдельной карточке картотеки или в отдельной записи может храниться всевозможная информация (например, фамилия, адрес, возраст,...). Эта информация называется **полем данных**. Объединение отдельных полей составляет логическое целое – **запись или предложение файла**. Прежде чем создавать или обрабатывать файлы в программе, следует позаботиться о том, чтобы нужная информация запоминалась в предложении файла.

Попробуем создать небольшой **адресный справочник**, в котором будут следующие поля данных:

- Имя (30 позиций)
- Фамилия (30 позиций)
- Улица (30 позиций)
- Почтовый индекс (6 позиций)
- Местность (30 позиций)
- Телефон (16 позиций)
- Заглавие (30 позиций)

Чтобы ввести данные в справочник, необходимо сначала **открыть файл данных**. Для этого используется оператор OPEN в следующем формате:

**OPEN Имя\_файла FOR Режим\_работы AS #Номер\_файла**

**Имя\_файла** – Имя файла с расширением и маршрутом.

**Режим\_работы** – Устанавливает разрешенный режим доступа к данным, хранящимся в файле.

**APPEND:** файл открывается для записи, причем новые предложения записываются в конец файла.

**INPUT:** файл открывается для чтения.

**OUTPUT:** файл открывается для записи, причем если файла не существует, то создается новый,

а если существует, то ранее существовавшая там информация будет уничтожена, а новая будет записана в начало.

**RANDOM:** файл состоит из записей фиксированной длины и открыт как для записи, так и для чтения.

**Номер\_файла:** Целое число между 1 и 255. Обращение к файлу из программы происходит под этим номером.

После того как файл обработан, с помощью команды CLOSE его необходимо **закрыть**. В программе это выглядит так:

**CLOSE #Номер\_файла**

Доступ к файлу возможен между командами OPEN и CLOSE, например, командой WRITE можно **записать предложение в файл**.

**WRITE Номер\_файла, Переменная [, Переменная]...**

Следующая программа показывает, как можно вводить данные для файла адресов. В режиме APPEND каждое записываемое предложение помещается в конец файла.

Таким образом, программу можно вызывать сколь угодно часто и дописывать данные в файл. Если же режим APPEND изменить на OUTPUT, данные будут каждый раз обновляться.

```

** Программа для ввода адресов **
'Объявление полей данных
DIM nachname$, vorname$, strasse$, plz$, ort$, telefon$, stichwort$
DIM nochein$ 'Переключатель
'Открыть файл в режиме APPEND
OPEN "adresse.asc" FOR APPEND AS #1
'Ввод данных будет продолжаться до тех пор, пока в поле nochein$
'будет вводиться "Y" или "y"
DO
CLS
'Ввести поля данных
INPUT "Фамилия : ", nachname$
INPUT "Имя . ", vorname$
INPUT "Улица : ", strasse$
INPUT "Почт.индекс: ", plz$
INPUT "Местность : ", ort$
INPUT "Телефон : ", telefon$
INPUT "Заглавие : ", stichwort$
'Записать предложение
WRITE #1,nachname$,vorname$,strasse$,plz$,ort$,telefon$,stichwort$
PRINT
'Спросить, продолжить ли ввод
INPUT "Следующий адрес? (Y/N)", nochein$
LOOP WHILE UCASE$(nochein$)="Y"
'Закрывать файл
CLOSE #1
'Конец программы

```

Далее необходимо вывести содержимое созданного файла на экран. Это значит, что данные последовательно читаются, и отдельные поля при необходимости появляются на экране. Для распознавания конца файла используется функция EOF.

## 2.Функция EOF

Функция *EOF* (*Номер\_файла*) применяется с аргументом *Номер\_файла* и возвращает логическое значение "истина", если достигнут конец файла, и "ложь" – в противном случае.

Отдельные записи файла считываются одним из вариантов команды INPUT. Ранее было рассмотрено считывание данных с клавиатуры и запоминание полученной информации в переменных. Для того чтобы **читать данные из файла**, следует применять оператор INPUT в таком виде:

```
INPUT #Номер_файла, Поле_данных_1 [,Поле_данных_2]... [,Поле_данных_n]
```

```

'Работа с файлами
'Программа для последовательного чтения адресов
'Объявление полей данных
DIM nachname$, vorname$, strasse$, plz$, ort$, telefon$,stichwort$
'Открыть файл в режиме INPUT
OPEN "adresse.asc FOR INPUT AS #1
CLS
'Читать файл до тех пор, пока функция EOF не примет
'значение "ложь" (0)
DO UNTIL EOF(1)
'0 до тех пор, пока не достигнут конец файла
'Читать запись

```

```

INPUT #1,nachname$,vorname$,strasse$,plz$,ort$,telefon$, stichwort$
'Вывести на экран
PRINT nachname$," "; vorname$; " "; strasse$
PRINT plz$; " "; ort$, " ", telefon$; " "; stichwort$
PRINT LOOP
'Закрыть файл CLOSE *1

```

### 3. Работа с адресами в справочнике

В данном разделе работу с адресами в справочнике дополним несколькими важными функциями. Прежде всего, должны быть разработаны удобные формы представления справочных данных на экране. Не следует обходить вниманием этот раздел, даже если Вас отпугнут несколько пространные примеры. Интенсивная проработка этого раздела, несомненно даст Вам ценный материал для собственной практики программирования. У Вас появится возможность повторить основные операции QBASIC.

Программа управления адресами должна выполнять следующие действия: искать адрес и изменять адрес. В главной программе для выбора отдельных действий необходимо создать меню. Из меню программа разветвляется на отдельные процедуры, которые выполняют независимые задачи. Преимущество такой структуры программы в том, что, во-первых, можно достаточно быстро получить доступ к любому фрагменту рабочей программы, а также меню - очень удобный интерфейс любой программы. Для начала попробуем нарисовать некую рамку, в которой и будет находиться наша программа. Пример программы представлен ниже.

```

CLS
FOR i = 17 TO 64
COLOR 1
LOCATE 10, i
PRINT "=" 'Прорисовка 1-й горизонтальной линии
LOCATE 23, i
PRINT "=" 'Прорисовка 2-й горизонтальной линии
NEXT i
FOR j = 11 TO 22
LOCATE j, 17
PRINT "" 'Прорисовка 1-й вертикальной линии
LOCATE j, 65
PRINT "|" 'Прорисовка 2-й вертикальной линии
NEXT j
LOCATE 10, 17 ' Прорисовка уголков
PRINT "?"
LOCATE 10, 65
PRINT "?"
LOCATE 23, 17
PRINT "?"
LOCATE 23, 64
PRINT "?"
COLOR 2
LOCATE 21, 39
PRINT "OK"

```

Теперь мы имеем рамочку в центре экрана и внутри написано ОК. Далее, во первых нам нужен обработчик клавиш для передвижения по меню, а во вторых требуется составить алгоритм прорисовки экрана после каждого нового перехода от одного пункта к другому.

Попробуйте составить программу самостоятельно.

Форма отчетности: отчёт по лабораторной работе выполнить в соответствии с методическими указаниями «Оформление текстовых документов» и включить в отчёт по лабораторному практикуму.

Основная литература [1,2,7].

Контрольные вопросы для самопроверки

1. Что такое массив данных?
2. Как получать статистических данных о массиве?

## ГРЕЧЕСКИЙ АЛФАВИТ

Название буквы	Строчная	Прописная	Название буквы	Строчная	Прописная	Название буквы	Строчная	Прописная
1.Альфа	$\alpha$	Α	9.Йота	$\iota$	Ι	17.Ро	$\rho$	Ρ
2.Бета	$\beta$	Β	10.Каппа	$\kappa$	Κ	18.Сигма	$\sigma$	Σ
3.Гамма	$\gamma$	Γ	11.Лямбда	$\lambda$	Λ	19.Тау	$\tau$	Τ
4.Дельта	$\delta$	Δ	12.Мю	$\mu$	Μ	20.Ипсилон	$\upsilon$	Υ
5.Эпсилон	$\epsilon$	Ε	13.Ню	$\nu$	Ν	21.Фи	$\phi$	Φ
6.Дзета	$\zeta$	Ζ	14.Кси	$\xi$	Ξ	22.Хи	$\chi$	Χ
7.Эта	$\eta$	Η	15.Омикрон	$\omicron$	Ο	23.Пси	$\psi$	Ψ
8.Тета	$\theta$	Θ	16.Пи	$\pi$	Π	24.Омега	$\omega$	Ω

### 9.2. Методические указания по выполнению контрольной работы

Контрольная работа выполняется в текстовом редакторе MS Word. Необходимо выполнить следующие задания.

1. Создать 3 абзаца различного стиля (в каждом абзаце не менее 20 строк связанного текста произвольной тематики).

1.1 Параметры 1-го абзаца: Arial, 14 пт., первая строка – отступ, выравнивание по левому краю, 2-ой межстрочный интервал.

1.2 Параметры 2-го абзаца: Arial, 10 пт., синий, курсив, первая строка - выступ, выравнивание по ширине страницы, 1,5-й межстрочный интервал.

1.3 Параметры 3-го абзаца: Times New Roman, 12 пт., черный, обычный, все строки с одной позиции, выравнивание по ширине страницы, отступы: слева - 2,5 см, справа - 4,5 см, одинарный межстрочный интервал.

2. Создать три списка различного типа (нумерованный, маркированный, многоуровневый).

3. Создать текст, с элементом форматирования в несколько колонок не менее половины страницы (газетный стиль).

4. Используя Microsoft Equation 3 создать 5 произвольных формул, содержащих индексы, специальные символы, символы греческого алфавита.

5. Во второй абзац текста вставить готовый рисунок из файла с режимом обтекания в тексте.

6. Создать собственный рисунок, используя панель инструментов «Рисование», редактора Word.

7. Создать таблицу средствами редактора Word (строк - не менее 8, столбцов - не менее 4). Заполнить данными и подвести итоговые вычисления.

8. Оформить фрагмент текста в виде объявления (рамка, заливка, ярлычки с номером телефона и т.д.).

9. Создать визитную карточку, используя средства WordArt на панели «Рисование».

10. В документе должны быть заголовки различного уровня (не менее двух уровней).

11. Распечатать документ, указав в верхнем колонтитуле: фамилию, группу, в нижнем колонтитуле: номера страниц.

12. Выполнить и распечатать титульный лист и лист содержания.

Сшить отчет сдать контрольную работу. При выполнении рекомендуется пользоваться методическими указаниями.

## 10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Microsoft Imagine Premium: Microsoft Windows Professional 7;

Microsoft Office 2007 Russian Academic OPEN No Level;

Антивирусное программное обеспечение Kaspersky Security;

Adobe Reader.

## 11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

<i>Вид занятия</i>	<i>Наименование аудитории</i>	<i>Перечень основного оборудования</i>	<i>№ ЛР</i>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Лк	Лекционная / семинарская аудитория	Учебная мебель	-
ЛР	Дисплейная аудитория	10-ПК: AMD Athlon™ 64x2 Dual Core Processor 5000+, 2,60 ГГц, ОЗУ 2,00Гб; Монитор Samsung 943N MY194S; Принтер: HP LaserJet 1160; Учебная мебель	№1 ... №15
кн	Читальный зал №1	10-ПК i5-2500/Н67/4Gb(мониторTFT19 Samsung); принтер HP LaserJet P2055D; Учебная мебель	-
СР	Читальный зал №1	10-ПК i5-2500/Н67/4Gb(мониторTFT19 Samsung); принтер HP LaserJet P2055D; Учебная мебель	-

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ  
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

**1. Описание фонда оценочных средств (паспорт)**

№ компетенции	Элемент компетенции	Раздел	Тема	ФОС
ОПК-1	способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно коммуникационных технологий и с учетом основных требований информационной безопасности	1. Общая характеристика процессов сбора, передачи, обработки и накопления информации.	<p><b>1.1.</b> Основные понятия информатики. Предметная область информатики. Структура и закономерности протекания информационных процессов.</p> <p><b>1.2.</b> Понятие информации. Виды информации. Подходы к оценке количества информации. Основы и методы защиты информации.</p> <p><b>1.3.</b> Единицы измерения и носители информации. Кодирование информации. Системы счисления (двоичная, восьмеричная, шестнадцатеричная).</p> <p><b>1.4.</b> Понятие файла. Каталоги и маршруты. Работа с файлами.</p>	экзаменационные вопросы 1.1-1.4
		2. Локальные и глобальные сети ЭВМ.	<p><b>2.1.</b> Сети ЭВМ (локальные, корпоративные, глобальные).</p> <p><b>2.2.</b> Мировая компьютерная сеть Интернет, принципы ее организации и работы.</p> <p><b>2.3.</b> Вредоносное программное обеспечение. Методы защиты данных.</p> <p><b>2.4.</b> Основы устройства ЭВМ. Поколения ЭВМ.</p>	экзаменационные вопросы 2.1-2.4
		3. Технические средства реализации информационных процессов	<p><b>3.1.</b> ПЭВМ, ее устройство и основные технические характеристики</p> <p><b>3.2.</b> Рабочие станции и серверы. Кластеры и мэйнфреймы.</p> <p><b>3.3.</b> Операционные си-</p>	экзаменационные вопросы 3.1-3.3

			стемы. Понятие операционной системы, ее назначение и функции.	
		4. Программные средства реализации информационных процессов.	4.1. Сервисное программное обеспечение. Работа с дисками.	экзаменационные вопросы 4.1-4.4
			4.2. Офисное программное обеспечение.	
			4.3. Программное обеспечение для работы с компьютерной графикой. Понятие векторной и растровой графики. Графические редакторы.	
			4.4. Системы управления базами данных	
		5. Алгоритмизация и программирование	5.1. Понятие алгоритма. Способы описания алгоритмов. Структурные схемы алгоритмов.	экзаменационные вопросы 5.1-5.4
			5.2. Языки программирования. Сравнительные характеристики языков программирования	
			5.3. Стили программирования. Процедурное программирование. Функциональное программирование. Логическое программирование.	
			5.4. Понятие об объектноориентированном программировании.	

## 2. Экзаменационные вопросы

№ п/п	Компетенции		ЭКЗАМЕНАЦИОННЫЕ ВОПРОСЫ	№ и наименование раздела
	Код	Определение		
1	2	3	4	5
1.	ОПК-1	способность к коммуникации в устной и письменной формах на русском и иностранном языках для решения задач межличностного и межкультур-	<p>1.1 Основные понятия информатики.</p> <p>1.2 Предметная область информатики.</p> <p>1.3 Структура и закономерности протекания информационных процессов.</p> <p>1.4 Понятие информации.</p>	1. Общая характеристика процессов сбора, передачи, обработки и накопления информации.

		ного взаимодействия	<p><b>1.5</b> Виды информации.  <b>1.6</b> Подходы к оценке количества информации.  <b>1.7</b> Основы и методы защиты информации.  <b>1.8</b> Единицы измерения и носители информации.  <b>1.9</b> Кодирование информации.  <b>1.10</b> Системы счисления (двоичная, восьмеричная, шестнадцатеричная).  <b>1.11</b> Структура ЭВМ.  <b>1.12</b> Понятие файла.  <b>1.13</b> Каталоги и маршруты.  <b>1.14</b> Работа с файлами.</p>	
			<p><b>2.1</b> Сети ЭВМ (локальные, корпоративные, глобальные).  <b>2.2</b> Мировая компьютерная сеть Интернет, принципы ее организации и работы.  <b>2.3</b> Вредоносное программное обеспечение. Методы защиты данных.</p>	<p><b>2.</b> Локальные и глобальные сети ЭВМ.</p>
			<p><b>3.1</b> Основы устройства ЭВМ. Поколения ЭВМ.  <b>3.2</b> ПЭВМ, ее устройство и основные технические характеристики  <b>3.3</b> Рабочие станции и серверы. Кластеры и мэйнфреймы.</p>	<p><b>3.</b> Технические средства реализации информационных процессов</p>
			<p><b>4.1</b> Операционные системы.  <b>4.2</b> Понятие операционной системы, ее назначение и функции.  <b>4.3</b> Сервисное программное обеспечение.  <b>4.4</b> Работа с дисками.  <b>4.5</b> Офисное программное обеспечение.  <b>4.6</b> Программное обеспечение для работы с компьютерной графикой.  <b>4.7</b> Понятие векторной и растровой графики.  <b>4.8</b> Графические редакторы.  <b>4.9</b> Системы управления базами данных.  <b>4.10</b> Microsoft Word. Основы работы.  <b>4.11</b> Microsoft Excel. Основы работы.  <b>4.12</b> Microsoft Power Point. Создание презентаций.  <b>4.13</b> Каталоги и маршруты.</p>	<p><b>4.</b> Программные средства реализации информационных процессов.</p>

			<p><b>4.14</b> Работа с файлами.</p> <p><b>5.1</b> Понятие алгоритма.</p> <p><b>5.2</b> Способы описания алгоритмов.</p> <p><b>5.3</b> Структурные схемы алгоритмов.</p> <p><b>5.4</b> Языки программирования.</p> <p><b>5.5</b> Сравнительные характеристики языков программирования.</p> <p><b>5.6</b> Стили программирования.</p> <p><b>5.7</b> Процедурное программирование.</p> <p><b>5.8</b> Функциональное программирование.</p> <p><b>5.9</b> Логическое программирование.</p> <p><b>5.10</b> Понятие об объектно-ориентированном программировании</p>	<p><b>5.</b> Алгоритмизация и программирование.</p>
--	--	--	--	---

### 3. Описание показателей и критериев оценивания компетенций

Показатели	Оценка	Критерии
<p><b>Знать:</b>  <b>ОПК-1</b> - методы и процессы сбора, передачи, обработки и накопления информации;</p> <ul style="list-style-type: none"> <li>- технические и программные средства реализации информационных процессов;</li> <li>- модели решения функциональных и вычислительных задач;</li> <li>- алгоритмизацию и программирование;</li> <li>- языки программирования;</li> <li>- базы данных;</li> <li>- программное обеспечение и технологию программирования;</li> <li>- компьютерную графику;</li> <li>- локальные сети и их использование при решении прикладных задач обработки данных;</li> </ul> <p><b>Уметь:</b>  <b>ОПК-1</b> использовать возможности вычислительной техники и программного обеспечения в отрасли</p>	<p><b>отлично</b></p>	<p>Знает методы и процессы сбора и обработки информации; владеет учебной лексикой, лексику повседневного пользования.</p> <p>Имеет точное знание о принципах работы компьютера, программировании, базах данных, компьютерной графике, локальных сетях.</p> <p>Умеет использовать вычислительную технику для решения практических и прикладных задач.</p> <p>Владеет вычислительными системами и системами программирования.</p>
	<p><b>хорошо</b></p>	<p>Знает особенности программирования, решения вычислительных задач, обработки информации.</p> <p>Умеет решать некоторые задачи средней сложности на компьютере.</p> <p>Владеет основными навыками работы на компьютере, свободно набирает текст.</p>
	<p><b>удовлетворительно</b></p>	<p>Знает и может частично ответить на вопросы, затрагивающие суть и возможности компьютерной техники; испытывает трудности решения некоторых задач; знает элементарные правила пользования клавиатурой и мышью.</p> <p>Умеет выполнять простые задания по обработке информации; испытывает трудности в поддержании диалога о возможностях часто используемых компьютерных программ; понимает основные положения; допускает серьезные ошибки при решении более сложных задач.</p> <p>Владеет умением пользования компьютерными средствами на посредственном уровне, допускает грубые ошибки при ответах на вопросы о</p>

<b>Владеть:</b> <b>ОПК-1</b> пользовательскими вычислительными системами и системами программирования.		часто применяемых способах выполнения команд.
	<b>неудовлетворительно</b>	Знаком частично с устройством и назначением частей компьютера. Умеет выполнять элементарные операции с мышью и клавиатурой. Не владеет навыками элементарных операций по управлению элементами рабочего стола, записи, копирования и удаления файлов.

#### 4. Тесты

##### Задание № 1

Фундаментальное свойство информации, означающее, что информация может менять способ и форму своего существования, – это ...

- воспроизводимость
- преобразуемость
- передаваемость
- стираемость

##### Задание № 2

В одном Гбайте содержится \_\_\_\_\_ Кбайт.

Известно, что

1 килобайт	= $2^{10}$ байтов	= 1024 байта
1 мегабайт	= $2^{20}$ байтов	= 1024 килобайта
1 гигабайт	= $2^{30}$ байтов	= 1024 мегабайта
1 терабайт	= $2^{40}$ байтов	= 1024 гигабайта

##### Задание № 3

Существует \_\_\_\_\_ различные(-ых) последовательности(-ей) из символов «А» и «В», длиной ровно в пять символов.

- 120
- 10
- 32
- 25

##### Задание № 4

Последняя цифра числа  $78965431267_{10}$  в двоичной системе счисления равна \_\_\_\_\_

##### Задание № 5

Логическое выражение  $\text{НЕ}((Y > 4) \text{ ИЛИ } (Y < 1)) \text{ И } (Y = 2)$  истинно, когда значение переменной  $Y$  равно ...

- 2
- 4
- 1
- 0

**Задание № 6**

Для запоминания 2 байт информации достаточно \_\_\_ триггера(ов).

- 1
- 2
- 8
- 16

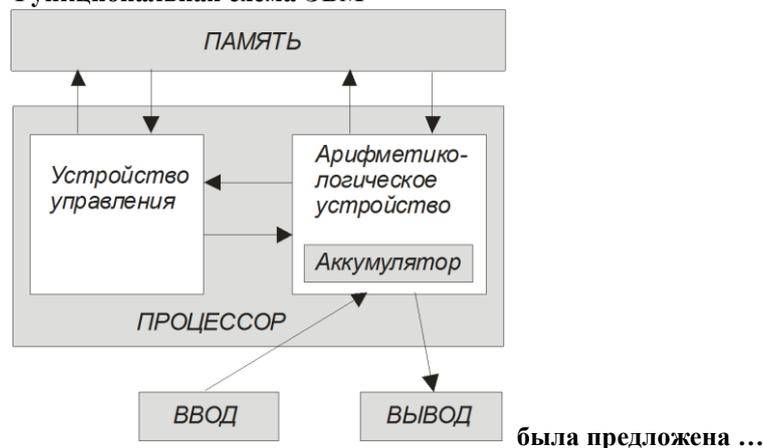
**Задание № 7**

Механическое устройство, позволяющее складывать числа, изобрел ...

- П. Нортон
- Б. Паскаль
- Г. Лейбниц
- Д. Нейман

**Задание № 8**

**Функциональная схема ЭВМ**



- Биллом Гейтсом
- Дж. фон Нейманом
- Норбертом Винером
- Готфридом Лейбницем

**Задание № 9**

**Верными являются утверждения, что ...**

Укажите не менее двух вариантов ответа

- манипулятор «Мышь» обладает системой команд
- оперативная память не обладает системой команд
- системой команд обладает процессор
- контроллеры устройств компьютера обладают системой команд

**Задание № 10**

**Оперативное запоминающее устройство (ОЗУ) относится к виду памяти ...**

- внутренней
- на магнитных дисках
- на оптических дисках
- внешней

**Задание № 11**

**При вводе символов с клавиатуры для переключения между режимами вставки и замены служит клавиша...**

- <Num Lock>
- <Shift>
- <PrtScr>
- <Insert>

**Задание № 12**

**При изучении объекта реальной действительности можно создать...**

- несколько различных видов моделей, каждая из которых отражает те или иные существенные признаки объекта
- только математическую модель
- только одну модель, отражающую совокупность признаков объекта
- точную копию объекта во всех проявлениях его свойств поведения

**Задание № 13**

**Образной информационной моделью является \_\_\_\_\_ автобуса.**

- расписание движения
- макет
- фотография
- карта с маршрутом

**Задание № 14**

**При составлении географической карты используется \_\_\_\_\_ моделирование.**

- компьютерное
- графическое
- сетевое
- математическое

**Задание № 15**

**При составлении модели расписания движения автобусов внутри республики полученная модель может рассматриваться как пример \_\_\_\_\_ модели.**

- графической
- компьютерной
- словесной
- табличной

**Задание № 16**

**Модемы бывают ...**

Укажите не менее двух вариантов ответа

- внешними
- внутренними
- матричными
- планшетными

**Задание № 17**

**Национальным (географическим) доменом не является ...**

- org
- ua
- ca
- uk

**Задание № 18**

**Доступ к файлу *net.edu*, находящемуся на сервере *news.com*, осуществляется по протоколу *ftp*. URL-адресом указанного файла является ...**

- ftp://news.com/net.edu
- ftp:news.com/net.edu
- ftp/news.com/net.edu
- ftp://net.edu/news.com

**Задание № 19**

**Программы, называемые сторожами, сканерами, детекторами, ревизорами, фильтрами, докторами, вакцинами, являются \_\_\_\_\_ программами.**

- игровыми
- виртуальными
- медицинскими
- антивирусными

### Задание № 20

Итоги чемпионата среди команд определяются следующим образом: за победу начисляется 3 очка, за ничью – 1 очко, за поражение очки не начисляются. При равенстве очков в турнирной таблице выше должна стоять команда, у которой лучше разность забитых и пропущенных мячей.

Введите в электронную таблицу исходные данные (слова можно сокращать).

	A	B	C	D	E	F	G	H
1	Итоги чемпионата							
2	Название команды	Выигрыши	Ничьи	Поражения	Мячей забито	Мячей пропущено	Разность	Очки
3	Сатурн	4	8	6	10	23		
4	Звезда	5	5	8	15	22		
5	Молот	8	5	5	22	20		
6	Дизель	10	1	7	25	24		
7	Горняк	10	4	4	27	19		
8	Шахтер	8	5	5	24	20		
9	Заря	8	6	4	25	19		
10	Луч	7	7	4	25	19		
11	Восток	8	6	4	26	22		
12	Волна	11	4	3	33	15		

Введите в электронную таблицу формулы для расчета значений в столбцах G и H.

По полученным результатам установите соответствие между командами и набранными очками:

- 1) Горняк
- 2) Дизель
- 3) Луч

Укажите соответствие для каждого пронумерованного элемента задания

- 29
- 28
- 34
- 31

### Задание № 21

Семантический аспект – это характеристика информации с точки зрения ее ...

- количества
- полезности
- качества
- смысла

### Задание № 22

В конкурсе участвовали 20 студентов, 8 школьников и 4 учащихся колледжа. Количество информации в сообщении о том, что победил школьник, считая, что победа любого из участников равновероятна, составит \_\_\_\_ бит(-а).

- 3
- 2
- 4
- 1

### Задание № 23

Растровый графический файл содержит черно-белое изображение с 2 градациями цвета (черный и белый) размером 800 × 600 точек. Определите необходимый для кодирования цвета точек (без учета служебной информации о формате, авторстве, способах сжатия и пр.) размер этого файла на диске в байтах.

- 480
- 480 000
- 3 840 000
- 60 000

### Задание № 24

Перевести двоичное число 1100101001101010111 в восьмеричную систему счисления.

**Задание № 25**

Для того чтобы логическое выражение

$$(X \& \neg X) ? (Y \& \neg Y)$$

было тождественно истинным, вместо знака ? в нем ...

- нельзя поставить знак дизъюнкции ( $\vee$ ), но можно поставить знак конъюнкции ( $\&$ )
- можно поставить знак дизъюнкции ( $\vee$ ), но не знак конъюнкции ( $\&$ )
- можно поставить как знак дизъюнкции ( $\vee$ ), так и знак конъюнкции ( $\&$ )
- нельзя поставить ни знак дизъюнкции ( $\vee$ ), ни знак конъюнкции ( $\&$ )

**Задание № 26**

Основной функциональной частью АЛУ является ...

- дизъюнктор
- конъюнктор
- инвертор
- сумматор

**Задание № 27**

Автором проекта первой автоматической вычислительной машины (Аналитической машины) является ...

- Чарльз Бэббидж
- Блез Паскаль
- Джон фон Нейман
- Михаил Васильевич Ломоносов

**Задание № 28**

Согласно классификации параллельных архитектур по Флинну ЭВМ, построенные по принципам фон Неймана, относят к типу ...

- SIMD - одиночный поток команд и множественный поток данных
- MIMD - множественный поток команд, множественный поток данных
- SISD - один поток команд, один поток данных
- MISD - множественный поток команд и одиночный поток данных

**Задание № 29**

К основным характеристикам центрального процессора относятся ...

Укажите не менее двух вариантов ответа

- количество слотов расширения
- разрядность
- объем встроенной кэш-памяти
- тактовая частота

**Задание № 30**

Промежуточный буфер с быстрым доступом, содержащий копию той информации, которая хранится в памяти с менее быстрым доступом, но с наибольшей вероятностью может быть оттуда запрошена, называют ...

- кэш-памятью
- внешней памятью
- памятью на оптических дисках
- памятью на магнитных дисках

**Задание № 31**

К устройствам координатного ввода данных относятся ...

- мышь, джойстик, трекбол
- мышь, принтер, аудиокolonки
- трекбол, дисплей, стриммер

дисплей, сканер, джойстик

**Задание № 32**

**Слово «модель» происходит от латинского «modulus», что в переводе означает ...**

- образец
- явление
- объект
- процесс

**Задание № 33**

**В классификации моделей по области использования не бывает моделей ...**

- опытных
- динамических
- игровых (ролевых)
- учебных

**Задание № 34**

**Основная область применения первого нейрокомпьютера, моделирующего структуру человеческого мозга, – это ...**

- выполнение сложных вычислений
- решение задач, требующих значительного количества вычислений
- синтезирование речи
- распознавание образов

**Задание № 35**

**При проведении занятий проектор включается в начале занятия и выключается через 10 минут после окончания занятия. Расписание занятий на текущий день представлено на рисунке.**

тема	начало	окончание
MS Word	8:00	9:20
Интернет	9:35	11:15
Поиск в сети Интернет	12:20	13:55
Сетевые сервисы	14:45	15:45

**Максимальное время отдыха проектора (с выключенным питанием) составит \_\_\_\_ минут.**

- 5
- 65
- 40
- 55

**Задание № 36**

**К основным компонентам вычислительных сетей относят ...**

Укажите не менее двух вариантов ответа

- антивирусные программы
- сетевое программное обеспечение
- компьютеры
- коммуникационное оборудование

**Задание № 37**

**Служба DNS (*Domain Name Service*) отвечает за ...**

- обеспечение маршрутизации коммутационных пакетов
- таблицы соответствия друг другу символьных и числовых адресов компьютеров, подключенных к сети Интернет
- пересылку электронных писем
- обеспечение устойчивости работы вычислительной сети

**Задание № 38**

**Доступ к файлу *http.txt*, находящемуся на сервере *txt.org* в папке *internet* осуществляется по протоколу *ftp*. URL-адресом этого файла является ...**

- http.txt://ftp/internet\txt.org
- ftp: \ \ txt.org \ internet \ http.txt
- ftp://txt.org/internet/http.txt
- ftp://txt.org/internet/http.txt

**Задание № 39**

**Антивирусные программы, выполняющие после запуска проверку заданной области файловой структуры компьютера, называются ...**

- антивирусные мониторы
- программы-вакцины
- программы-брандмауэры
- антивирусные сканеры

**Задание № 40**

Студенты выполняют 5 тестов по информатике. За каждый тест можно получить от 0 до 10 баллов. Если за тест № 3 получено не менее 6 баллов, то этот результат увеличивается на 20 %. Если суммарное количество полученных при тестировании баллов меньше 20, то это соответствует оценке «2»; оценке «3» соответствует количество баллов от 20 до 29; оценке «4» – от 30 до 39; оценке «5» – 40 баллов и выше.

Введите в электронную таблицу исходные данные (слова можно сокращать).

	A	B	C	D	E	F	G	H
1	<b>Итоги тестирования</b>							
2								
3	<b>ФИО</b>	<b>Тест 1</b>	<b>Тест 2</b>	<b>Тест 3</b>	<b>Тест 4</b>	<b>Тест 5</b>	<b>Сумма баллов</b>	<b>Оценка</b>
4	Авилова О.С.	5	4	5	3	2		
5	Арбузов Н.В.	5	6	4	5	4		
6	Бондаренко Д.А.	8	7	8	7	8		
7	Васильева К.А.	9	7	7	10	9		
8	Голубев В.В.	5	6	6	4	4		
9	Денисов А.М.	8	9	9	8	9		
10	Игнатъев С.А	6	6	7	8	7		
11	Кудинов В.С.	6	6	5	6	6		
12	Серова Т.В.	8	6	7	9	8		
13	Филатова М.Е.	5	6	7	6	5		
14								
15	<b>Средний результат</b>							

Введите в электронную таблицу формулы для расчета:

- значений в столбцах G и H (используйте логическую функцию «ЕСЛИ»);
- среднего значения в ячейке G15.

По полученным расчетам установите соответствие между следующими участниками олимпиады и количеством набранных ими баллов:

- Арбузов Н. В.
- Игнатъев С. А.
- Серова Т. В.

Укажите соответствие для каждого нумерованного элемента задания

- 39,4
- 35,4
- 24
- $F = \overline{(A \& B) \vee (B \vee C)} \& (\overline{A} \& C)$
- $F = \overline{(A \& B) \vee (B \& C)} \vee (\overline{A} \& C)$
- $F = (A \& B) \vee (B \& C) \vee (\overline{A} \& C)$
- $F = \overline{(A \& B) \vee (B \& C)} \vee (A \& C)$

**Задание № 47**

Расположите прообразы современных вычислительных устройств, появившиеся в механический период, в правильной последовательности.

Укажите порядковый номер для всех вариантов ответов

- суммирующая машина Блеза Паскаля
- разностная машина Чарльза Бэббиджа
- устройство, позволяющее выполнять все четыре арифметических действия, Готфрида Лейбница
- Аналитическая машина Чарльза Бэббиджа

Задание № 48

Архитектура процессора, основанная на концепции «более компактные и простые инструкции выполняются быстрее», – это \_\_\_\_\_ архитектура.

- конвейерная
- 8-ми разрядная
- CISC-
- RISC-

Задание № 49

К базовой конфигурации персонального компьютера относится ...

Укажите не менее двух вариантов ответа

- принтер
- системный блок
- монитор
- клавиатура

Задание № 50

В USB флеш-накопителях (флеш-картах) используется ...

- электронная энергонезависимая перезаписываемая память
- магнитная карта
- небольшой прямоугольный DVD R/W
- память на магнитных сердечниках

Задание	1	2	3	4	5	6	7	8	9	10
Ответ	2	1048576	3	1	2	16	2	2	2,3,4	1
Задание	11	12	13	14	15	16	17	18	19	20
Ответ	4	1	3	2	4	1,2	1	3	4	
Задание	21	22	23	24	25	26	27	28	29	30
Ответ	4	2	4	1451527 <sub>8</sub>	4	4	1	3	2,3,4	1
Задание	31	32	33	34	35	36	37	38	39	40
Ответ	1	1	2	4	4	2,3,4	2	3	4	
Задание	41	42	43	44	45	46	47	48	49	50
Ответ	1	3	1	4	1	1	1,3,2,4	4	2,3,4	1

Оценка	Критерии оценки тестов
зачтено	оценка «зачтено» выставляется, если все задания выполнены без существенных ошибок, показано умение иллюстрировать теоретические положения конкретными примерами, продемонстрировано усвоение ранее изученного материала, устойчивость используемых умений и навыков, что свидетельствует о сформированности компетенции. Допускаются незначительные ошибки.
не зачтено	оценка «не зачтено» выставляется, если обнаружено незнание или непонимание большей или наиболее важной части учебного материала; не сформированы компетенции, умения и навыки. Многие требования, предъявляемые к заданию, не выполнены.

### Критерии оценки устного ответа

	«Удовлетворительно»	«Хорошо»	«Отлично»
<b>Знание материала</b>	Не раскрыто основное содержание темы.	Не полно раскрыто содержание темы, но показано общее понимание темы.	Представленное содержание темы раскрыто в полном объеме и соответствует теме.
<b>Владение речью и терминологией</b>	допущены ошибки в изложении темы, затрудняющие понимание сказанного. Не может привести примеры, демонстрирует полное неумение делать обобщение, выводы, сравнения.	в изложении темы допущены ошибки, не искажающие общего понимания; приведение примеров вызывает затруднение, обобщение, выводы, сравнение делаются с помощью преподавателя.	тема изложена грамотным языком, с точным использованием лексики. В определении понятий показано умение иллюстрировать материал конкретными примерами; делать обобщение, выводы, сравнение.
<b>Степень самостоятельности</b>	Содержание материала излагалось с многочисленными подсказками, показавшими незнание темы	содержание материала излагалось с помощью наводящих вопросов и подсказок.	содержание материала изложено самостоятельно, без наводящих вопросов.

#### 5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности.

Дисциплина «Информатика» направлена на получение теоретических знаний и практических навыков и умений для активного применения в повседневной жизни и профессиональной деятельности.

Изучение дисциплины «Информатика» предусматривает:

- лекции
- выполнение лабораторных работ;
- выполнение курсовой работы;
- зачет, экзамен;
- самостоятельную работу обучающихся.

В ходе освоения раздела 1 «Общая характеристика процессов сбора, передачи и обработки информации» студенты должны уяснить, что такое информация и в каком виде она представлена в компьютере.

В разделе 2 «Локальные и глобальные сети» внимание следует обратить на смысл современного компьютера, как единицы кокай-либо сети.

В разделе 3 «Технические средства реализации информационных процессов» следует осознать стремительное увеличение параметров компьютера, его быстродействия и объема памяти.

В разделе 4 «Програмные средства реализации информационных процессов» главное состоит в осознании и использования компьютера для решения различных задач, стоящих перед человеком.

В разделе 5 «Алгоритмизация и программирование» даётся понятие о специфике составления программ, алгоритме и качественного изменения способа решения задач программиста.

Необходимо овладеть навыками и умениями применения изученных методов для решения широкого класса различных задач студента и применения знаний и навыков при реализации проектов в течение всей последующей деятельности.

В процессе изучения дисциплины рекомендуется на первом этапе обратить внимание на освоении клавиатуры, работы с мышью и повышения скорости выполнения элементарных операций

Овладение ключевыми понятиями является на основе понимания современного компьютера как системы, состоящей из трёх основных частей: аппаратная часть, программная часть и человек.

При подготовке к зачету/экзамену рекомендуется особое внимание уделить следующим вопросам: какие функции выполняют отдельные составляющие аппаратной части компьютера и для предназначена та или иная компьютерная программа.

В процессе проведения лабораторных работ происходит закрепление знаний, формирование умений и навыков использования компьютера для выполнения офисных работ.

Самостоятельную работу необходимо начинать с повторения материала, полученного во время проведения аудиторных занятий. Далее студент расширяет объём своих знаний, читая учебную литературу и активно изучая возможности компьютерных программ. Необходимо акцентировать внимание на операциях, наиболее часто применяемых пользователями компьютеров.

В процессе консультации с преподавателем студент проявляя уже полученные знания получает возможность освоить более трудный для понимания материал.

Работа с литературой является важнейшим элементом в получении знаний по дисциплине. Прежде всего, необходимо воспользоваться списком рекомендуемой по данной дисциплине литературой. Дополнительные сведения по изучаемым темам можно найти в библиотеке и Интернете.

Предусмотрено проведение аудиторных занятий с применением интерактивных форм (проектная деятельность). Внеаудиторная работа предполагает самостоятельную работу обучающихся на своих компьютерах с целью закрепления полученных на занятиях знаний, приобретения умений и навыков.

## АННОТАЦИЯ рабочей программы дисциплины

### Информатика

#### 1. Цель и задачи дисциплины

Целью изучения дисциплины является ознакомление с методами, способами получения, хранения и обработки информации; получение практических навыков в использовании современного прикладного программного обеспечения и возможностей глобальных компьютерных сетей.

Задачами изучения дисциплины являются: использования информационных технологий при проектировании и разработке в составе коллектива исполнителей новых видов транспорта и транспортного оборудования, а также транспортных изделий; эффективного использования материалов, оборудования, соответствующих алгоритмов и программ расчетов параметров технологических процессов; информационного поиска и анализа информации по объектам исследований; участия в составе коллектива исполнителей в обосновании и применении новых информационных технологий.

#### 2. Структура дисциплины

2.1. Распределение трудоемкости по отдельным видам учебных занятий, включая самостоятельную работу: лекций – 8 часов, лабораторных занятий – 8 часа, самостоятельная работа обучающихся – 119 часов.

Общая трудоемкость дисциплины составляет 144 часа, 4 зачетных единицы.

2.2. Основные разделы дисциплины:

- 1 – Общая характеристика процессов сбора, передачи, обработки и накопления информации.
- 2 – Локальные и глобальные сети ЭВМ.
- 3 – Технические средства реализации информационных процессов.
- 4 – Программные средства реализации информационных процессов.
- 5 – Алгоритмизация и программирование.

#### 3. Планируемые результаты обучения (перечень компетенций)

Процесс изучения дисциплины направлен на формирование следующих компетенций:

**ОПК-1** способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

4. Вид промежуточной аттестации: экзамен.

*Протокол о дополнениях и изменениях в рабочей программе  
на 20\_\_-20\_\_ учебный год*

1. В рабочую программу по дисциплине вносятся следующие дополнения:

\_\_\_\_\_

\_\_\_\_\_

2. В рабочую программу по дисциплине вносятся следующие изменения:

\_\_\_\_\_

\_\_\_\_\_

Протокол заседания кафедры № \_\_\_\_\_ от «\_\_» \_\_\_\_\_ 20\_\_ г.,  
(разработчик)

Заведующий кафедрой \_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(Ф.И.О.)

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки: 23.03.03 «Эксплуатация транспортно-технологических машин и комплексов» от «14» декабря 2015 года № 1470

для набора 2018 года и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «12» марта 2018 г. № 130.

**Программу составил:**

Слепенко Е.А., доцент, кандидат технических наук \_\_\_\_\_

Рабочая программа рассмотрена и утверждена на заседании кафедры МиТ

от «11» декабря 2018 г., протокол № 6

и.о.заведующего кафедрой МиТ \_\_\_\_\_ Е.А. Слепенко

СОГЛАСОВАНО:

И.о.заведующего выпускающей кафедрой \_\_\_\_\_ Е.А. Слепенко

Директор библиотеки \_\_\_\_\_ Т.Ф. Сотник

Рабочая программа одобрена методической комиссией механического факультета

от « 14 » декабря 2018 г., протокол № 4.

Председатель методической комиссии факультета \_\_\_\_\_ Г.Н. Плеханов

СОГЛАСОВАНО:

Начальник

учебно-методического управления \_\_\_\_\_ Г.П. Нежевец

Регистрационный № \_\_\_\_\_