

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.06

НАПРАВЛЕНИЕ ПОДГОТОВКИ

**15.03.05 Конструкторско-технологическое обеспечение
машиностроительных производств**

ПРОФИЛЬ ПОДГОТОВКИ

Технология машиностроения

Программа прикладного бакалавриата

Квалификация (степень) выпускника: бакалавр

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	4
3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ.....	4
3.1 Распределение объёма дисциплины по формам обучения.....	4
3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости	4
4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	5
4.1 Распределение разделов дисциплины по видам учебных занятий	5
4.2 Содержание дисциплины, структурированное по разделам и темам	5
4.3 Лабораторные работы.....	70
4.4 Семинары / практические занятия.....	70
4.5 Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат.....	70
5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ	71
6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ	72
7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	72
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	73
9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....	73
9.1. Методические указания для обучающихся по выполнению лабораторных работ	73
9.2. Методические указания по выполнению курсового проекта (курсовой работы), контрольной работы, РГР, реферата	77
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	78
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	78
Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	79
Приложение 2. Аннотация рабочей программы дисциплины	82
Приложение 3. Протокол о дополнениях и изменениях в рабочей программе	83
Приложение 4. Фонд оценочных средств для текущего контроля успеваемости по дисциплине.....	84

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

Вид деятельности выпускника

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

Цель и задачи дисциплины

Целью изучения дисциплины является:

- сформировать знания, умения и навыки, необходимые для рационального использования средств современных информационно-коммуникационных технологий при решении задач, связанных с обработкой информации, ее поиском, систематизацией, сохранением, представлением, передачей;

- ознакомить студентов с ролью новых информационно-коммуникационных технологий в современном производстве, науке, повседневной практике, с перспективами развития компьютерной техники;

Задачами изучения дисциплины являются:

- получение представления об информатике (предмет, задачи, методы и сущность информатики) и освоение базовых понятий о средствах современной вычислительной техники и принципы работы технических и программных средств;

- ознакомление с алгоритмами и языком программирования, как средством реализации алгоритма на ЭВМ;

- изучение основных принципов программирования;

- ознакомление с пакетами прикладных программ, применяемых для обработки данных

- приобретение навыков в использование современной компьютерной техники для сбора, хранения и обработки информации;

- приобретение навыков работы на ЭВМ.

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ОПК-2	- 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>знать:</p> <ul style="list-style-type: none"> - основы информационной и библиографической культуры, основные информационно-коммуникационные технологии и основные требования информационной безопасности; <p>уметь:</p> <ul style="list-style-type: none"> - решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; <p>владеть:</p> <ul style="list-style-type: none"> - способами применения информационно-коммуникационных технологий для решения стандартных профессиональных задач.
ОПК-3	- способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности	<p>знать:</p> <ul style="list-style-type: none"> - современные информационные технологии, прикладные программные средства используемые при решении задач профессиональной деятельности; <p>уметь:</p> <ul style="list-style-type: none"> - использовать современные информационные технологии, прикладные программные средства используемые при решении задач профессиональной деятельности; <p>владеть:</p> <ul style="list-style-type: none"> - способностью использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.Б.06 «Информатика» относится к базовой части.

Дисциплина «Информатика» базируется на знаниях, полученных при изучении такой учебной дисциплины, как «Математика».

Основываясь на изучении соответствующих дисциплин, «Информатика» представляет основу для изучения дисциплин:

- «Технология создания инженерных программ»;
- «Технология поточного автоматизированного производства»;
- «Теория автоматического управления»;
- «Управление системами и процессами»;
- «Автоматизация производственных процессов в машиностроении»;
- «Средства и методы автоматизации производства».

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 (проект), контрольная работа, реферат, РГР	Вид промежуточной аттестации
			Всего часов (с экз.)	Аудиторных часов	Лекции	Лабораторные работы	Семинары / Практические занятия	Самостоятельная работа		
Очная	1	1,2	180	85	34	51	-	68	-	Зачет, Экзамен
Заочная	-	-	-	-	-	-	-	-	-	-
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 семестрам, час	
			1	2
I. Контактная работа обучающихся с преподавателем (всего)	85	34	51	34
Лекции (Лк)	34	34	17	17
Лабораторные работы (ЛР)	51	-	34	17
Групповые (индивидуальные) консультации	+	-	+	+
II. Самостоятельная работа обучающихся (СР)	68	-	21	47
Подготовка к лабораторным работам	44	-	14	30
Подготовка к экзамену в течение семестра	17	-	-	17
Подготовка к зачету	7	-	7	-
III. Промежуточная аттестация	27	-	-	27
Экзамен				
зачет	+	-	+	-
Общая трудоемкость дисциплины час.	180	-	72	108
зач. ед.	5	-	2	3

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Распределение разделов дисциплины по видам учебных занятий

- для очной 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.	Теоретическая информатика. Основы программирования.	72	17	34	21
1.1.	Информатика и информация.	5	3	-	2
1.2.	Структура ЭВМ. Поколения ЭВМ.	4	2	-	2
1.3.	Алгоритм	4	2	-	2
1.4.	Язык программирования QBAS-IC	59	10	34	15
2.	Программное обеспечение ЭВМ. Интернет.	81	17	17	47
2.1.	Операционные системы.	6	2	-	4
2.2.	Текстовый редактор.	28	6	8	14
2.3.	Табличный процессор.	30	6	9	15
2.4.	Архиваторы. Антивирусы.	5	1	-	4
2.5.	Интернет.	12	2	-	10
	ИТОГО	153	34	51	68

4.2. Содержание дисциплины, структурированное по разделам и темам

Раздел 1. Теоретическая информатика. Основы программирования.

Тема 1.1. Информатика и информация (лекция – дискуссия 3 часа).

Мы начинаем изучать предмет, который носит название "Информатика".

Информатика - это наука, сложившаяся сравнительно недавно. Ее развитие связано с появлением в середине XX века электронно-вычислительных машин, которые явились мощными универсальными средствами для хранения, обработки и передачи информации.

Истоки информатики можно искать в глубине веков. Много столетий тому назад потребность выразить и запомнить информацию привела к появлению речи, письменности, счета. Люди пытались изобретать, а затем совершенствовать способы хранения, обработки и распространения информации. До сих пор сохранились свидетельства попыток наших далеких предков сохранять информацию - примитивные наскальные рисунки, записи на берестяной коре и глиняных дощечках, затем рукописные книги.

Появление в XVI веке печатного станка позволило значительно увеличить возможности человека обрабатывать нужные сведения. Это явилось важным этапом развития человечества. Хранение информации в печатном виде стало основным способом хранения информации и продолжало им оставаться вплоть до середины XX века. Только с появлением ЭВМ возникли принципиально новые, гораздо более эффективные способы хранения информации (рис. 1.1).

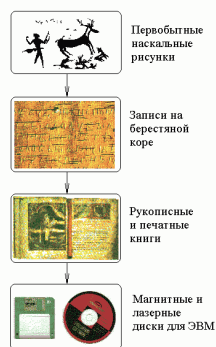


Рис. 1.1. Развитие способов хранения информации

Развивались способы передачи информации. Примитивный способ передачи посланий от человека к человеку сменился более прогрессивной почтовой связью. Почтовая связь давала достаточно надежный способ обмена информацией. Однако не следует забывать, что таким образом могли передаваться только сообщения, написанные на бумаге. А главное - скорость передачи сообщения была соизмерима только со скоростью передвижения человека. Изобретение телеграфа, телефона дало принципиально новые возможности обработки и передачи информации.

Появление электронно-вычислительных машин позволило обрабатывать, а в последствии и передавать информацию со скоростью, в несколько миллионов раз превышающей скорость обработки и передачи информации человеком (рис. 1.2, 1.3).

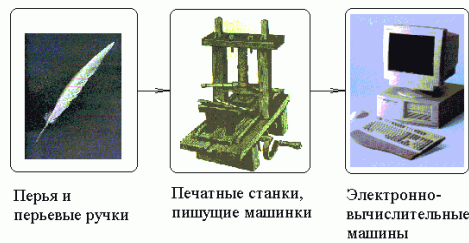


Рис. 1.2. Развитие способов обработки информации

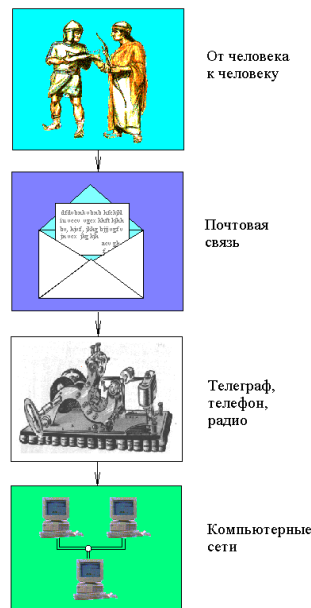


Рис. 1.3. Развитие способов передачи информации

Электронно-вычислительные машины и системы являются базой информатики. Поэтому информатика находит широкое применение в различных областях современной жизни: производстве, науке, образовании и других.

Приведем примеры. Ежедневно мы получаем прогноз погоды, даже не задумываясь каким образом он получен. На самом деле задача прогноза погоды является очень сложной и трудоемкой. Чтобы ее решить, необходимо собрать, а затем проанализировать информацию, поступающую с метеорологических станций, спутников.

Для этого применяются методы математического моделирования процессов в атмосфере и океане, решаются сложные системы уравнений. Потом полученные результаты представляются в удобной для человека форме (рис. 1.4). Все эти этапы невозможно произвести, не пользуясь методами и средствами информатики. А тем более, не применяя ЭВМ.



Рис. 1.4. Карта погоды Гидрометцентра России

Развитие современной науки предполагает проведение сложных и дорогостоящих экспериментов, таких как, например, при разработке термоядерных реакторов. Информатика позволяет заменить реальные эксперименты машинными. Это экономит колоссальные ресурсы, дает возможность обработать полученные результаты самыми современными методами. Кроме того, такие эксперименты занимают гораздо меньше времени, чем настоящие. А в некоторых областях науки, например астрофизике, проведение реального эксперимента просто невозможно. Здесь, в основном, все исследования проводятся посредством вычислительных экспериментов.

Можно приводить множество примеров использования информатики в различных сферах жизнедеятельности. Рассмотрим лишь еще один пример. Много лет, а можно сказать и веков, обучение происходило при непосредственном общении преподавателя с учащимся. В дальнейшем появилась заочная форма обучения, которая позволяла изучать необходимый материал самостоятельно. Но и она предполагала, в конечном итоге, сдачу экзаменов непосредственно преподавателю. Так что принципиальных новшеств в области получения образования не наблюдалось очень давно. Изменения стали наблюдаться только с появлением компьютеров. Первые попытки использовать компьютер в обучении были произведены в семидесятых годах двадцатого столетия и были не достаточно успешными. Это объясняется невысокой производительностью технических и программных средств того времени. Кроме того программы не позволяли учитывать индивидуальных особенностей обучающегося. Обучающие программы сегодняшнего поколения предлагают пользователю множество вариантов настройки. А это значит, что учащийся при освоении учебного материала сам устанавливает такие вещи, как скорость изучения, объем материала, степень сложности курса.

Многочисленные исследования говорят о том, что метод обучения при помощи компьютеров гораздо эффективнее старых традиционных методов. Сравнить эти методы сложно, однако доказано, что внимание при работе с компьютерной программой обучения усиливается, а поэтому время занятий сокращается примерно на 30%.

Уже давно учеными выявлена связь между способом, с помощью которого осваивается материал и способностью восстановить этот материал в памяти. Так вот, при прослушивании лекций только четверть услышанного материала остается в памяти. Если этот же материал воспринимать зрительно, то в памяти остается до трети полученных сведений. Применяя комбинированные способы обучения, доля усвоенного материала достигает половины. И только когда учащийся принимает активное участие в процессе изучения материала с помощью компьютерных программ, только тогда (как говорят последние научные исследования) доля усвоенного материала достигает 75%.

Следует отметить, что обучающие программы существуют не только для обучения в учебных заведениях, но и для обучения и переподготовки персонала различных фирм, а также для проведения квалификационных испытаний.

Дальнейшее развитие информатики, как и любой другой науки, влечет за собой новые достижения, открытия. А, следовательно, и новые области применения, которые, может быть трудно сегодня предположить.

Виды информации

Понятие **информации** является основополагающим понятием информатики. Любая деятельность человека представляет собой процесс сбора и переработки информации, принятия на ее основе решения и их выполнения. С появлением современных средств вычислительной техники информация стала выступать в качестве одного из важнейших ресурсов научно-технического прогресса.

Информация содержится в человеческой речи, текстах книг, журналов и газет, сообщениях радио и телевидения, показаниях приборов и т. д. Человек воспринимает информацию с помощью органов чувств, хранит и перерабатывает ее с помощью мозга и центральной нервной системы. Передаваемая информация обычно касается каких-то предметов или нас самих и связана с событиями, происходящими в окружающем нас мире.

В рамках науки информация является первичным и неопределяемым понятием. Оно предполагает наличие материального носителя информации, источника информации, передатчика информации, приемника и канала связи между источником и приемником. Понятие информации используется во всех сферах: науке, технике, культуре, социологии и повседневной жизни. Конкретное толкование элементов, связанных с понятием информации, зависит от метода конкретной науки, цели исследования или просто от наших представлений.

Термин **«информация»** происходит от латинского *informatio* — разъяснение, изложение, осведомленность. Энциклопедический словарь (М.: Сов. энциклопедия, 1990) определяет информацию в исторической эволюции: первоначально — сведения, передаваемые людьми устным, письменным или другим способом (с помощью условных сигналов, технических средств и т. д.); с середины XX века — общенаучное понятие, включающее обмен сведениями между людьми, человеком и автоматом, обмен сигналами в животном и растительном мире (передача признаков от клетки к клетке, от организма к организму).

Более узкое определение дается в технике, где это понятие включает в себя все сведения, являющиеся объектом хранения, передачи и преобразования.

Наиболее общее определение имеет место в философии, где под информацией понимается отражение реального мира. Информацию как философскую категорию рассматривают как один из атрибутов материи, отражающий ее структуру.

В эволюционном ряду *вещество -> энергия -> информация* каждое последующее проявление материи отличается от предыдущего тем, что людям было труднее его распознать, выделить и использовать в чистом виде. Именно сложность выделения различных проявлений материи обусловила, наконец, указанную последовательность познания природы человечеством.

Единица измерения информации

Понятие количества информации

Количеством информации называют числовую характеристику сигнала, отражающую ту *степень неопределенности* (неполноту знаний), которая исчезает после получения сообщения в виде данного сигнала. Эту меру неопределенности в теории информации называют **энтропией**. Если в результате получения сообщения достигается полная ясность в каком-то вопросе, говорят, что была получена полная или исчерпывающая информация и необходимости в получении дополнительной информации нет. И, наоборот, если после получения сообщения неопределенность осталась прежней, значит, информации получено не было (нулевая информация).

Приведенные рассуждения показывают, что между понятиями информации, неопределенность и возможность выбора существует тесная связь. Так, любая неопределенность предполагает возможность выбора, а любая информация, уменьшая неопределенность, уменьшает и возможность выбора. При полной информации выбора нет. Частичная информация уменьшает число вариантов выбора, сокращая тем самым неопределенность.

Пример. Человек бросает монету и наблюдает, какой стороной она упадет. Обе стороны монеты равноправны, поэтому одинаково вероятно, что выпадет одна или другая сторона. Такой ситуации приписывается начальная неопределенность, характеризуемая двумя возможностями. После того, как монета упадет, достигается полная ясность и неопределенность исчезает (становится равной нулю).

Приведенный пример относится к группе событий, применительно к которым может быть поставлен вопрос типа «да-нет». Количество информации, которое можно получить при ответе на вопрос типа «да-нет», называется **битом** (англ. *bit* — сокращенное от *binary digit* — двоичная единица). Бит — минимальная единица количества информации, ибо получить информацию меньшую, чем 1 бит, невозможно. При получении информации в 1 бит неопределенность уменьшается в 2 раза. Таким образом, каждое бросание монеты дает нам информацию в 1 бит.

В качестве других моделей получения такого же количества информации могут выступать электрическая лампочка, двухпозиционный выключатель, магнитный сердечник, диод и т. п. Включенное состояние этих объектов обычно обозначают цифрой 1, а выключенное — цифрой 0.

Рассмотрим систему из двух электрических лампочек, которые независимо друг от друга могут быть включены или выключены. Для такой системы возможны следующие состояния:

Лампа А 0 0 1 1

Лампа В 0 1 0 1

Чтобы получить полную информацию о состоянии системы, необходимо задать два вопроса типа «да-нет» — по лампочке А и лампочке В соответственно. В этом случае количество информации, содержащейся в данной системе, определяется уже в 2 бита, а число возможных состояний системы — 4. Если взять три лампочки, то необходимо задать уже три вопроса и получить 3 бита информации. Количество состояний такой системы равно 8 и т. д.

Связь между количеством информации и числом состояний системы устанавливается формулой Хартли:

$$i = \log_2 N,$$

где i — количество информации в битах; N — число возможных состояний. Ту же формулу можно представить иначе:

$$N = 2^i.$$

Группа из 8 битов информации называется **байтом**. Если бит — минимальная единица информации, то байт ее основная единица. Существуют производные единицы информации: килобайт (кбайт, кб), мегабайт (Мбайт, Мб) и гигабайт (Гбайт, Гб).

$$1 \text{ кб} = 1024 \text{ байта} = 2^{10} (1024) \text{ байтов.}$$

$$1 \text{ Мб} = 1024 \text{ кбайта} = 2^{20} (1024 \times 1024) \text{ байтов.}$$

$$1 \text{ Гб} = 1024 \text{ Мбайта} = 2^{30} (1024 \times 1024 \times 1024) \text{ байтов.}$$

Эти единицы чаще всего используют для указания объема памяти ЭВМ.

Бит — минимальная единица измерения информации (от английского *Binary digit* — двоичная цифра). Каждый бит может принимать значение 0 или 1. Битом также называют разряд ячейки памяти ЭВМ. Для измерения объема хранимой информации используются следующие единицы:

Измерения в байтах					
Десятичная приставка			Двоичная приставка		
Название	Символ	Степень	Название	Символ	Степень
килобайт	кВ	10^3	кибибайт	KiB	2^{10}
мегабайт	МВ	10^6	мебибайт	MiB	2^{20}
гигабайт	ГВ	10^9	гибибайт	GiB	2^{30}
терабайт	ТВ	10^{12}	тебибайт	TiB	2^{40}

петабайт	PB	10^{15}	пебибайт	PiB	2^{50}
эксабайт	EB	10^{18}	эксбибайт	EiB	2^{60}
зеттабайт	ZB	10^{21}	зебибайт	ZiB	2^{70}
йоттабайт	YB	10^{24}	йобибайт	YiB	2^{80}

Гигабайт — кратная единица измерения количества информации, равная 1 073 741 824 (2^{30}) стандартным (8-битным) байтам или 1 024 мегабайтам.

Приставка СИ *giga-* используется ошибочно, так как она обозначает умножение на 10^9 . Для 2^{30} же следует употреблять двоичную приставку *gibi-*. Сложившимся положением пользуются крупные корпорации, производящие жёсткие диски, которые при маркировке своих изделий под мегабайтом понимают 1 000 000 байт, а под гигабайтом — 1 000 000 000 байт.

Сигнал. Виды сигнала

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

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

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

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

Различают две формы представления информации — непрерывную и дискретную. Поскольку носителями информации являются сигналы, то в качестве последних могут использоваться физические процессы различной природы. Например, процесс протекания электрического тока в цепи, процесс механического перемещения тела, процесс распространения света и т. д. Информация представляется (отражается) значением одного или нескольких параметров физического процесса (сигнала), либо комбинацией нескольких параметров.

Сигнал называется непрерывным, если его параметр в заданных пределах может принимать любые промежуточные значения. Сигнал называется дискретным, если его параметр в заданных пределах может принимать отдельные фиксированные значения.

Следует различать непрерывность или дискретность сигнала по уровню и во времени.

На рис. 1.1 в виде графиков изображены: а) непрерывный по уровню и во времени сигнал X_{nn} ; б) дискретный по уровню и непрерывный во времени сигнал $X_{нд}$; в) непрерывный по уровню и дискретный во времени сигнал $X_{нд}$; г) дискретный по уровню и во времени сигнал $X_{дд}$.

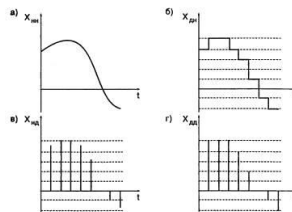


Рис. 1.1. Виды информационных процессов

Наконец, все многообразие окружающей нас информации можно сгруппировать по различным признакам, т. е. классифицировать по видам. Например, в зависимости от области возникновения информацию, отражающую процессы и явления неодушевленной природы, называют элементарной, процессы животного и растительного мира — биологической, человеческого общества — социальной.

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

Системы счисления

Системой счисления называется способ изображения чисел с помощью ограниченного набора символов, имеющих определенные количественные значения. Систему счисления образует совокупность правил и приемов представления чисел с помощью набора знаков (цифр).

Различают позиционные и непозиционные системы счисления. В позиционных системах каждая цифра числа имеет определенный вес, зависящий от позиции цифры в последовательности, изображающей число. Позиция цифры называется разрядом. В позиционной системе счисления любое число можно представить в виде:

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

$$A_n = \sum_{i=-k}^{m-1} a_i \cdot N^i, \quad (2.1)$$

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).

Пример 2.1. $A_{10}=37.25$.

В соответствии с формулой (2.1) 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 аппаратурной 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,...,9};

шестнадцатеричная - {0,1,2, ...,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.

Пример 2.2. Число в двоичной, восьмеричной и шестнадцатеричной системах счисления имеет следующее представление:

$$A_2 = 1100100,101;$$

$$A_8 = 144,5;$$

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

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

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

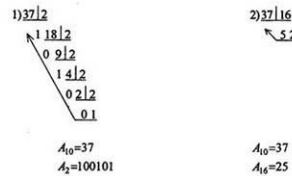
$$A_{16} = 6 \cdot 16^1 + 4 \cdot 16^0 + 10 \cdot 16^{-1}.$$

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.

Перевод целых чисел

Целое число с основанием N_1 переводится в систему счисления с основанием N_2 путем последовательного деления числа A_{n1} , на основании 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 .

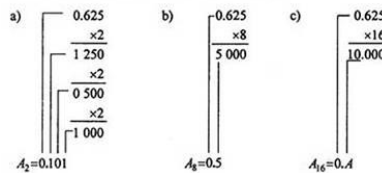
Пример 2.3. $A_{10}=37$ $A_2=?$ $A_{16}=?$



Перевод дробных чисел

Дробное число с основанием N_1 переводится в систему счисления с основанием N_2 путем последовательного умножения A_{n1} 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_{n1} в системе счисления N_2 .

Пример 2.4. $A_{10}=0,625$ $A_2=?$ $A_4=?$ $A_{16}=?$



Так как двоичная, восьмеричная и шестнадцатеричная системы связаны через степени числа 2, то преобразования между ними можно выполнять другим более простым способом. Для перевода из шестнадцатеричной (восьмеричной) системы счисления в двоичную достаточно двоичным кодом записать шестнадцатеричные коды цифр тетрадами (по 4 двоичных разряда) и триадами (по 3 двоичных разряда) - для восьмеричных цифр. Обратный перевод из двоичного кода производится в обратном порядке: двоичное число разбивается влево и вправо от границы целой и дробной частей на тетрады - для последующей записи цифр в шестнадцатеричном представлении, на триады - для записи их значений восьмеричными цифрами.

Таблица 2.1

Таблица кодирования текстовой информации ASCII

Radix : Hex		
0	1	!
2	3	"
4	5	#
6	7	\$
8	9	%
A	B	&
C	D	'
E	F	(
10	11)
12	13	*
14	15	+
16	17	,
18	19	<
1A	1B	=
1C	1D	>
1E	1F	?
20	21	@
22	23	A
24	25	B
26	27	C
28	29	D
2A	2B	E
2C	2D	F
2E	2F	G
30	31	H
32	33	I
34	35	J
36	37	K
38	39	L
3A	3B	M
3C	3D	N
3E	3F	O
40	41	P
42	43	Q
44	45	R
46	47	S
48	49	T
4A	4B	U
4C	4D	V
4E	4F	W
50	51	X
52	53	Y
54	55	Z
56	57	[
58	59	\
5A	5B]
5C	5D	^
5E	5F	_
60	61	`
62	63	a
64	65	b
66	67	c
68	69	d
6A	6B	e
6C	6D	f
6E	6F	g
70	71	h
72	73	i
74	75	j
76	77	k
78	79	l
7A	7B	m
7C	7D	n
7E	7F	o
80	81	p
82	83	q
84	85	r
86	87	s
88	89	t
8A	8B	u
8C	8D	v
8E	8F	w
90	91	x
92	93	y
94	95	z
96	97	{
98	99	
9A	9B	}
9C	9D	~
9E	9F	
AA	AB	A
AC	AD	B
AE	AF	C
B0	B1	D
B2	B3	E
B4	B5	F
B6	B7	G
B8	B9	H
BA	BB	I
BC	BD	J
BE	BF	K
C0	C1	L
C2	C3	M
C4	C5	N
C6	C7	O
C8	C9	P
CA	CB	Q
CC	CD	R
CE	CF	S
D0	D1	T
D2	D3	U
D4	D5	V
D6	D7	W
D8	D9	X
DA	DB	Y
DC	DD	Z
DE	DF	[
E0	E1	\
E2	E3]
E4	E5	^
E6	E7	_
E8	E9	`
EA	EB	a
EC	ED	b
EE	EF	c
EA	EB	d
EC	ED	e
EE	EF	f
F0	F1	g
F2	F3	h
F4	F5	i
F6	F7	j
F8	F9	k
FA	FB	l
FC	FD	m
FE	FF	n

Стандарт ASCII

Расширение ASCII

Первая половина таблицы стандартизована. Она содержит управляющие коды (от 00h до 20h и 77h). Эти коды из таблицы изъяты, так как они не относятся к текстовым элементам. Здесь же размещаются знаки пунктуации и математические знаки: 21h - !, 26h - &, 28h - (, 2Bh - +, ..., большие и малые латинские буквы: 41h - A, 61h - a,...

Вторая половина таблицы содержит национальные шрифты, символы псевдографики, из которых могут быть построены таблицы, специальные математические знаки. Нижнюю часть таблицы кодировок можно заменять, используя соответствующие драйверы - управляющие вспомогательные программы. Этот прием позволяет применять несколько шрифтов и их гарнитур.

Дисплей по каждому коду символа должен вывести на экран изображение символа - не просто цифровой код, а соответствующую ему картинку, так как каждый символ имеет свою форму.

Описание формы каждого символа хранится в специальной памяти дисплея - знакогенераторе.

Высвечивание символа на экране дисплея IBM PC осуществляется с помощью точек, образующих символьную матрицу.

Каждый пиксел в такой матрице является элементом изображения и может быть ярким или темным. Темная точка кодируется цифрой 0, светлая (яркая) - 1.

Если изображать в матричном поле знака темные пикселы точкой, а светлые - звездочкой, то можно графически изобразить форму символа.

Кодирование аудиоинформации - процесс более сложный. Аудиоинформация является аналоговой. Для преобразования ее в цифровую форму используют аппаратные средства: аналого-цифровые преобразователи (АЦП), в результате работы которых аналоговый сигнал оцифровывается — представляется в виде числовой последовательности. Для вывода оцифрованного звука на аудиоустройства необходимо проводить обратное преобразование, которое осуществляется с помощью цифро-аналоговых преобразователей (ЦАП).

Тема 1.2. Структура ЭВМ. Поколения ЭВМ (лекция – дискуссия 2 часа)

Структура ЭВМ

Более чем за полвека развития вычислительных средств прогресс в аппаратной реализации ЭВМ и их технических характеристик превзошел все прогнозы, и пока не заметно снижение его темпов. Несмотря на то, что современные ЭВМ внешне не имеют ничего общего с первыми моделями, основополагающие идеи, заложенные в них и связанные с понятием алгоритма, разработанным Аланом Тьюрингом, а также архитектурной реализацией, предложенной Джоном фон Нейманом, пока не претерпели коренных изменений (за исключением систем параллельной обработки информации).

Любая *ЭВМ неймановской архитектуры* содержит следующие основные устройства:

- арифметико-логическое устройство (АЛУ);
- устройство управления (УУ)
- запоминающее устройство (ЗУ);
- устройства ввода-вывода (УВВ);
- пульт управления (ПУ).

В современных ЭВМ АЛУ и УУ объединены в общее устройство, называемое центральным процессором. Обобщенная логическая структура ЭВМ представлена на рис. 1.3.

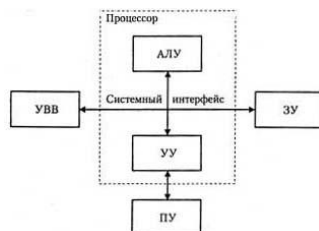


Рис. 1.3. Обобщенная логическая структура ЭВМ

Процессор, или **микروпроцессор**, является основным устройством ЭВМ. Он предназначен для выполнения вычисления по хранящейся в запоминающем устройстве программе и обеспечения общего управления ЭВМ. Быстродействие ЭВМ в значительной мере определяется скоростью работы процессора. Для ее увеличения процессор использует собственную память небольшого объема, именуемую местной или сверхоперативной, что в некоторых случаях исключает необходимость обращения к запоминающему устройству ЭВМ.

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

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

Внутренняя, или **основная память** — это запоминающее устройство, напрямую связанное с процессором и предназначенное для хранения выполняемых программ и данных, непосредственно участвующих в вычислениях. Обращение к внутренней памяти ЭВМ осуществляется с высоким быстродействием, но она имеет ограниченный объем, определяемый системой адресации машины.

Внутренняя память, в свою очередь, делится на **оперативную (ОЗУ)** и **постоянную (ПЗУ) память**. **Оперативная память**, по объему составляющая большую часть внутренней памяти, служит для приема, хранения и выдачи информации. При выключении питания ЭВМ содержимое оперативной памяти в большинстве случаев теряется. **Постоянная память** обеспечивает хранение и выдачу информации. В отличие от содержимого оперативной памяти, содержимое постоянной заполняется при изготовлении ЭВМ и не может быть изменено в обычных условиях эксплуатации. В постоянной памяти хранятся часто используемые (универсальные) программы, и данные, к примеру, некоторые программы операционной системы, программы тестирования оборудования ЭВМ и др. При выключении питания содержимое постоянной памяти сохраняется.

Внешняя память (ВЗУ) предназначена для размещения больших объемов информации и обмена ею с оперативной памятью. Для построения внешней памяти используются энергонезависимые носители информации (диски и ленты), которые к тому же являются переносимыми. Емкость этой памяти практически не имеет ограничений, а для обращения к ней требуется больше времени, чем к внутренней.

Внешние запоминающие устройства конструктивно отделены от центральных устройств ЭВМ (процессора и внутренней памяти), имеют собственное управление и выполняют запросы процессора без его непосредственного вмешательства. В качестве ВЗУ используют накопители на магнитных и оптических дисках, а также накопители на магнитных лентах.

ВЗУ по принципам функционирования разделяются на **устройства прямого доступа** (накопители на магнитных и оптических дисках) и **устройства последовательного доступа** (накопители на магнитных лентах). Устройства прямого доступа обладают большим быстродействием, поэтому они являются основными внешними запоминающими устройствами, постоянно используемыми в процессе функционирования ЭВМ. Устройства последовательного доступа используются в основном для резервирования информации.

Устройства ввода-вывода служат соответственно для ввода информации в ЭВМ и вывода из нее, а также для обеспечения общения пользователя с машиной. Процессы ввода-вывода протекают с использованием внутренней памяти ЭВМ. Иногда устройства ввода-вывода называют **периферийными** или **внешними устройствами** ЭВМ. К ним относятся, в частности, дисплеи (мониторы), клавиатура, манипуляторы типа «мышь», алфавитно-цифровые печатающие устройства (принтеры), графопостроители, сканеры и др. Для управления внешними устройствами (в том числе и ВЗУ) и согласования их с системным интерфейсом служат групповые устройства управления внешними устройствами, адаптеры или контроллеры.

Системный интерфейс — это конструктивная часть ЭВМ, предназначенная для взаимодействия ее устройств и обмена информацией между ними.

В больших, средних и супер-ЭВМ в качестве системного интерфейса используются сложные устройства, имеющие встроенные процессоры ввода-вывода, именуемые каналами. Такие устройства обеспечивают высокую скорость обмена данными между компонентами ЭВМ.

Отличительной особенностью малых ЭВМ является использование в качестве системного интерфейса системных шин. Различают ЭВМ с **многосишной структурой** и с **общей шиной**. В первых для обмена информацией между устройствами используются отдельные группы шин, во втором случае все устройства ЭВМ объединяются с помощью одной группы шин, в которую входят подмножества шин для передачи данных, адреса и управляющих сигналов. При такой организации системы шин обмен информацией между процессором, памятью и периферийными устройствами выполняется по единому правилу, что упрощает взаимодействие устройств машины.

Пульт управления служит для выполнения оператором ЭВМ или системным программистом системных операций в ходе управления вычислительным процессом. Кроме того, при техническом обслуживании ЭВМ за пультом управления работает инженерно-технический персонал. Пульт управления конструктивно часто выполняется вместе с центральным процессором.

Основные характеристики вычислительной техники

К основным характеристикам вычислительной техники относятся ее эксплуатационно-технические характеристики, такие, как быстродействие, емкость памяти, точность вычислений и др.

Быстродействие ЭВМ рассматривается в двух аспектах. С одной стороны, оно характеризуется количеством элементарных операций, выполняемых центральным процессором в секунду. Под элементарной операцией понимается любая простейшая операция типа сложения, пересылки, сравнения и т. д. С другой стороны, быстродействие ЭВМ существенно зависит от организации ее памяти. Время, затрачиваемое на поиск необходимой информации в памяти, заметно сокращается на быстродействии ЭВМ.

В зависимости от области применения выпускаются ЭВМ с быстродействием от нескольких сотен тысяч до миллиардов операций в секунду. Для решения сложных задач возможно объединение нескольких ЭВМ в единый вычислительный комплекс с требуемым суммарным быстродействием.

- Наряду с быстродействием часто пользуются понятием **производительность**. Если первое обусловлено, главным образом, использованием в ЭВМ системой элементов, то второе связано с ее архитектурой и разновидностями решаемых задач.

Емкость, или **объем памяти** определяется максимальным количеством информации, которое можно разместить в памяти ЭВМ. Обычно емкость памяти измеряется в байтах. Как уже отмечалось, память ЭВМ подразделяется на внутреннюю и внешнюю. Внутренняя, или оперативная память, по своему объему у различных классов машин различна и определяется системой адресации ЭВМ. Емкость внешней памяти из-за блочной структуры и съемных конструкций накопителей практически неограничена.

Точность вычислений зависит от количества разрядов, используемых для представления одного числа. Современные ЭВМ комплектуются 32- или 64-разрядными микропроцессорами, что вполне достаточно для обеспечения высокой точности расчетов самых разнообразных приложений. Однако, если этого мало, можно использовать удвоенную или утроенную разрядную сетку.

Система команд — это перечень команд, которые способен выполнить процессор ЭВМ. Система команд устанавливает, какие конкретно операции может выполнять процессор, сколько операндов требуется указать в команде, какой вид (формат) должна иметь команда для ее распознавания. Количество основных разновидностей команд невелико, с их помощью ЭВМ способны выполнять операции сложения,

вычитания, умножена деления, сравнения, записи в память, передачи числа из регистра в регистр, преобразования из одной системы счисления в другую и т. д. При необходимости выполняете модификация команд, учитывающая специфику вычислений. Обычно в ЭВМ используется от десятков до сотен команд (с учетом их модификации).

На современном этапе развития вычислительной техники используются два основных подхода при формировании системы команд процессора. С одной стороны, это традиционный подход, связанный с разработкой процессоров с полным набором команд, — архитектура **CIS** (Complete Instruction Set Computer — компьютер с полным набором команд). С другой стороны, это реализация в ЭВМ сокращенного набора простейших, но часто употребляемых команд, что позволяет упростить аппаратные средства процессора и повысить его быстродействие — архитектура **RISC** (Reduced Instruction Set Computer — компьютер с сокращенным набором команд).

Стоимость ЭВМ зависит от множества факторов, в частности от быстродействия, емкости памяти, системы команд и т. д. Большое влияние на стоимость оказывает конкретная комплектация ЭВМ и, в первую очередь, внешние устройства, входящие в состав машины. Наконец, стоимость программного обеспечения ощутимо влияет на стоимость ЭВМ.

Надежность ЭВМ — это способность машины сохранять свои свойства при заданных условиях эксплуатации в течение определенного промежутка времени. Количественной оценкой надежности ЭВМ, содержащей элементы, отказ которых приводит к отказу всей машины, могут служить следующие показатели:

- вероятность безотказной работы за определенное время при данных условиях эксплуатации;
- наработка ЭВМ на отказ;
- среднее время восстановления машины и др.

Для более сложных структур типа вычислительного комплекса или системы понятие «отказ» не имеет смысла. В таких системах отказы отдельных элементов приводят к некоторому снижению эффективности функционирования, а не к полной потере работоспособности в целом.

Важное значение имеют и другие характеристики вычислительной техники, например: универсальность, программная совместимость, вес, габариты, энергопотребление и др. Они принимаются во внимание при оценивании конкретных сфер применения ЭВМ.

Перспективы развития вычислительных средств

Появление новых поколений ЭВМ обусловлено расширением сферы их применения, требующей более производительной, дешевой и надежной вычислительной техники. В настоящее время стремление к реализации новых потребительских свойств ЭВМ стимулирует работы по созданию машин пятого и последующего поколений. Вычислительные средства пятого поколения, кроме более высокой производительности и надежности при более низкой стоимости, обеспечиваемых новейшими электронными технологиями, должны удовлетворять качественно новым *функциональным требованиям*:

- работать с базами знаний в различных предметных областях и организовывать на их основе системы искусственного интеллекта;
- обеспечивать простоту применения ЭВМ путем реализации эффективных систем ввода-вывода информации голосом, диалоговой обработки информации с использованием естественных языков, устройств распознавания речи и изображения;
- упрощать процесс создания программных средств путем автоматизации синтеза программ.

В настоящее время ведутся интенсивные работы как по созданию ЭВМ пятого поколения традиционной (неймановской) архитектуры, так и по созданию и апробации перспективных архитектур и схемотехнических решений. На формальном и прикладном уровнях исследуются архитектуры на основе параллельных абстрактных вычислителей (матричные и клеточные процессоры, систолические структуры, однородные вычислительные структуры, нейронные сети и др.) Развитие вычислительной техники с высоким параллелизмом во многом определяется элементной базой, степенью развития параллельного программного обеспечения и методологией распараллеливания алгоритмов решаемых задач.

Проблема создания эффективных систем *параллельного программирования*, ориентированных на высокоуровневое распараллеливание алгоритмов вычисления и обработки данных, представляется достаточно сложной и предполагает дифференцированный подход с учетом сложности распараллеливания и необходимости синхронизации процессов во времени.

Наряду с развитием архитектурных и схемотехнических решений ведутся работы по совершенствованию *технологий производства интегральных схем* и по созданию принципиально новых *элементных баз*, основанных на оптоэлектронных и оптических принципах.

В плане создания принципиально новых архитектур вычислительных средств большое внимание уделяется проектам *нейрокомпьютеров*, базирующихся на понятии нейронной сети (структуры на формальных нейронах), моделирующей основные свойства реальных нейронов. В случае применения био- или оптоэлементов могут быть созданы соответственно *биологические* или *оптические* нейрокомпьютеры. Многие исследователи считают, что в следующем веке нейрокомпьютеры в значительной степени вытеснят современные ЭВМ, используемые для решения трудно формализуемых задач. Последние достижения в микроэлектронике и разработка элементной базы на основе биотехнологий дают возможность прогнозировать создание биокомпьютеров.

Важным направлением развития вычислительных средств пятого и последующих поколений является *интеллектуализация ЭВМ*, связанная с наделением ее элементами интеллекта, интеллектуализацией интерфейса с пользователем и др. Работа в данном направлении, затрагивая, в первую очередь, программное обеспечение, потребует и создания ЭВМ определенной архитектуры, используемых в системах управления базами знаний, — *компьютеров баз знаний*, а так же других подклассов ЭВМ. При этом ЭВМ должна обладать способностью к обучению, производить ассоциативную обработку информации и вести интеллектуальный диалог при решении конкретных задач.

В заключение отметим, что ряд названных вопросов реализован в перспективных ЭВМ пятого поколения либо находится в стадии технической проработки, другие — в стадии теоретических исследований и поисков.

Первое поколение ЭВМ

Идея использования программного управления для построения устройства, автоматически выполняющего арифметические вычисления, была впервые высказана английским математиком У.Бэббиджем еще в 1833г. Однако его попытки построить механическое вычислительное устройство с программным управлением не увенчались успехом.

Первой работающей универсальной автоматически управляемой ВМ считается расчетно-механическая машина "Марк - 1" (США, 1944г.). Простой машины составляли большую часть времени. Столь же низкая производительность оказалась и у машины "Марк - 2", построенной на реле улучшенной конструкции.

Проект первой ЭВМ ЭНИАК был разработан Дж.Моугли (США, 1942г.); в 1946г машина вступила в строй. В этой машине 18.000 электрических ламп, 1500 электромеханических реле. Применение ламп повысило скорость выполнения операций в 1000 раз по сравнению с устройством "Марк - 1".

За точку отсчета эры ЭВМ принимают сеансы опытной эксплуатации машины ЭНИАК, которые начались в Пенсильванском университете в 1946г. (общий вес – 30т, производительность - 5000 операций в секунду). Спустя 40 лет после пуска первой ЭВМ ежегодное производство компонентов ВТ оценивалось к 1985г. в 10^{14} активных логических элементов (active elements groups), что эквивалентно 1 ЭНИАК на каждого жителя земли. Для сравнения: за 500 лет развития книгопечатания к 1962г. общий тираж всех изданий достиг уровня 2 книги на каждого жителя Земли.

Электронные лампы стали элементной базой ВМ первого поколения. Основная схема – симметричный триггер был создан в 1918г. советским ученым Бонч-Бруевичем М.А. В 1919г. аналогичная схема была разработана также американскими учеными Икклзом и Джорданом.

Первые проекты отечественных ЭВМ были предложены С.А. Лебедевым, Б.И. Рамеевым в 1948г. В 1949-51гг. по проекту С.А. Лебедева была построена МЭСМ (малая электронно-счетная машина). К ЭВМ 1-го поколения относится и БЭСМ-1 (большая электронно-счетная машина), разработка которой под руководством С.А. Лебедева была закончена в 1952г., она содержала 5 тыс. ламп, работала без сбоев в течение 10 часов. Быстродействие достигало 10 тыс. операций в секунду. Почти одновременно проектировалась ЭВМ "Стрела" под руководством Ю.Я. Базилевского, в 1953г. она была запущена в производство. Позже появилась ЭВМ "Урал - 1", положившая начало большой серии машин "Урал", разработанных и внедренных в производство под руководством Б.И. Рамеева. В 1958г. запущена в серийное производство ЭВМ первого поколения М – 20 (быстродействие до 20 тыс. операций/с).

Второе поколение ЭВМ

С появлением транзисторов в середине 50-х годов на смену первого поколения ЭВМ пришли ЭВМ 2-го поколения, построенные на полупроводниковых приборах.

В нашей стране были созданы полупроводниковые ЭВМ разных назначений: малые ЭВМ серий "Наири" и "Мир", средние ЭВМ со скоростью работы 5-30 тыс. операций/с – "Минск - 22" и "Минск – 32", "Раздан – 2", "Раздан – 3", БЭСМ – 4, М – 220 и лучшая из машин второго поколения – БЭСМ – 6 со скоростью работы до 1 млн. опер/с.

Третье поколение ЭВМ

В начале 60-х годов возникло новое направление в электронике – интегральная электроника. Использование интегральных схем для построения ЭВМ стало революцией в ВТ и способствовало появлению машин 3-го поколения.

С 1972г. начался выпуск моделей первой очереди ЕС ЭВМ (совместно с социалистическими странами). Ряд – 1 : ЕС – 1010, 1020, 1022, 1030, 1033, 1040, 1050, 1052. Вторая очередь (Ряд - 2) : ЕС – 1015, 1025, 1035, 1045, 1055, 1060, 1065 имела более современную схемотехническую, конструкторско-технологическую базу, за счет чего у них увеличилась производительность, и расширились функциональные возможности.

Четвёртое поколение ЭВМ

Одна из характерных особенностей ЭВМ 4-го поколения - переход от интегральных функциональных схем к интегральным подсистемам ЭВМ. Подсчитано, что внедрение БИС увеличивает надежность не менее чем в 10 раз. Из отечественных ЭВМ к машинам 4-го поколения, прежде всего, относятся машины семейства "Эльбрус".

Таблица 1.1 показывает связь между основными параметрами схемотехники и поколениями ЭВМ. Быстродействие характеризуется задержкой распространения сигнала, вносимой одним элементарным элементом (конъюнктом, дизъюнктом и т. д.). Важный показатель – плотность упаковки, количество единиц элементов, приходящихся на 1см³.

Признак, параметр ЭВМ	Поколения					
	1-ое 1946-1955	2-ое 1955-1965	3-е 1965-1970	после 70г.	4-ое 80г.	после
Основные элементы	Реле, электронные лампы	Полупроводниковые приборы	ИС	БИС	СБИС	
Быстродействие (задержка/элемент или схема)	1мс	1мкс	10нс	1нс	< 1нс	
Плотность упаковки, эл-тов/см ³	0,1	2-3	10-20	1000	> 10000	

Три этапа информационной технологии: эволюция критериев.

В 1953г. создатель теории информации американский математик Клод Шеннон писал: "Наши ВМ выглядят как ученые-схоласты. При вычислении длинной цепи арифметических операций ЦВМ значительно обгоняют человека. Когда же пытаются приспособить ЦВМ для выполнения неарифметических операций, они оказываются неуклюжими и неприспособленными для такой работы."

1 Этап: машинные ресурсы. Отмеченные Шенноном функциональные ограничения, а также устрашающая стоимость первых ЭВМ полностью определяют основную задачу информационной технологии 50-х – начала 60-х гг. - повышение эффективности обработки данных по уже формализованным или легко формализуемым алгоритмам.

Основной целью тогда было – уменьшить общее число машинных тактов, которых требовала для своего решения та или иная программа, а также объем занимаемой ею ОЗУ. Основные затраты на обработку данных находились тогда почти в прямой зависимости от затраченного на них машинного времени.

2 Этап: программирование. В середине 60-х годов начался 2-й этап развития информационной технологии, который продолжался до начала 80-х годов. От технологии эффективного исполнения программ к технологии эффективного программирования – так можно было определить общее направление смены критериев эффективности в течение этого этапа. Наиболее известным результатом этого первого радикального пересмотра критериев технологии программирования стала созданная в начале 70-х годов ОС UNIX. Операционную систему UNIX, нацеленную, прежде всего, на повышение эффективности труда программистов, разработали сотрудники "Белл Лэбс" К. Томпсон и Д. Ритчи, которых совершенно не удовлетворяли имеющиеся примитивные средства проектирования программ, ориентированные на пакетный режим. На рубеже 80-х годов UNIX рассматривалась как классический образец ОС – она начала триумфальное шествие на мини-ЭВМ серии PDP – 11 в середине 70-х годов.

3 Этап: формализация знаний. Персональный компьютер, как правило, имеет развитые средства самообучения пользователя-новичка работе за пультом, гибкие средства защиты от его ошибок и, самое главное, все аппаратно-программные средства такой ЭВМ подчинены одной "сверхзадаче" - обеспечить "дружественную реакцию" машины на любые, в том числе неадекватные, действия пользователя. Основная задача персональных вычислений - формализация профессиональных знаний – выполняемая, как правило, самостоятельно непрограммирующим пользователем или при минимальной технической поддержке программиста.

Тема 1.3. Алгоритм (лекция – дискуссия 2 часа)

Само слово «алгоритм» возникло из названия латинского перевода книги арабского математика IX века Аль-Хорезми «Algoritmi de numero Indorum», что можно перевести как «Трактат Аль-Хорезми об арифметическом искусстве индусов». Составление алгоритмов и вопросы их существования являются предметом серьёзных математических исследований.

Свойства алгоритма. При составлении и записи алгоритма необходимо обеспечить, чтобы он обладал рядом свойств.

Однозначность алгоритма, под которой понимается единственность толкования исполнителем правила построения действий и порядок их выполнения. Чтобы алгоритм обладал этим свойством, он должен быть записан командами из системы команд исполнителя.

Конечность алгоритма – обязательность завершения каждого из действий, составляющих алгоритм, и завершенность выполнения алгоритма в целом.

Результативность алгоритма, предполагающая, что выполнение алгоритма должно завершиться получением определённых результатов.

Массовость, т. е. возможность применения данного алгоритма для решения целого класса задач, отвечающих общей постановке задачи. Для того чтобы алгоритм обладал свойством массовости, следует составлять алгоритм, используя обозначения величин и избегая конкретных значений.

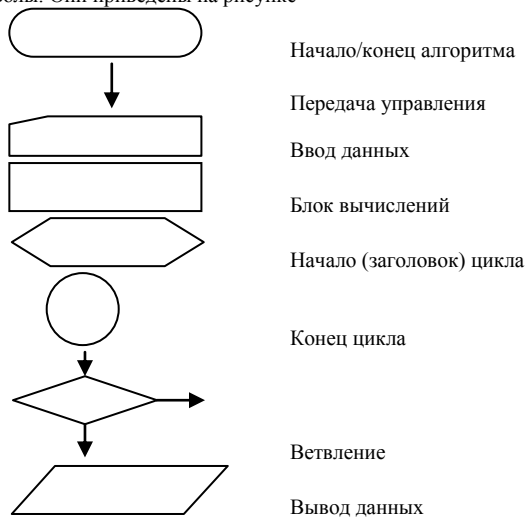
Правильность алгоритма, под которой понимается способность алгоритма давать правильные результаты решения поставленных задач.

Эффективность – для решения задачи должны использоваться ограниченные ресурсы компьютера (процессорное время, объём оперативной памяти и т. д.).



Представление алгоритма в виде блок-схемы

Для разработки структуры программы удобнее пользоваться записью алгоритма в виде *блок-схемы* (в англоязычной литературе используется термин *flow-chart*). Для изображения основных алгоритмических структур и блоков на блок-схемах используют специальные графические символы. Они приведены на рисунке

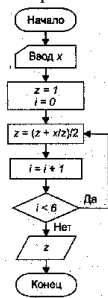


Составим алгоритм вычисления квадратного корня из произвольного положительного вещественного числа x методом Герона и запишем его на естественном языке, а также в виде блок-схемы. Метод основан на многократном применении формулы:

$$z_{n+1} = \frac{1}{2} \left(z_n + \frac{x}{z_n} \right)$$

при
 $z_0 = 1$.

Числовая последовательность z_n в пределе при $n \rightarrow \infty$ сходится к искомому значению. Выполним только 5 итераций метода, считая, что при этом будет достигнута достаточно хорошая точность. Обычно десяти итераций метода Герона более чем достаточно для достижения хорошей точности расчёта. Оба варианта записи алгоритма:



1. Ввести x .
2. Присвоить $z = 1$.
3. Присвоить $i = 0$.
4. Присвоить $z = (z + x/z)/2$.
5. Присвоить $i = i + 1$.
6. Если $i < 6$, то перейти к шагу 4, иначе напечатать значение z .

А теперь займёмся самым любимым занятием школьников всех времён и народов – решением квадратного уравнения:

$$ax^2 + bx + c = 0.$$

Будем полагать, что коэффициенты этого уравнения a , b и c представляют собой вещественные числа. Простейший случай предполагает, что все коэффициенты отличны от нуля. В зависимости от знака дискриминанта квадратного уравнения

$$D = b^2 - 4ac$$

возможны три случая:

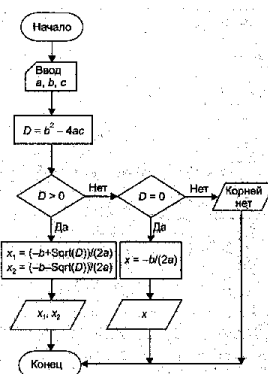
1. Если $D > 0$, то имеются два различных вещественных корня, которые можно вычислить по следующим формулам:

$$x_1 = \frac{-b - \sqrt{D}}{2a}, \quad x_2 = \frac{-b + \sqrt{D}}{2a}.$$

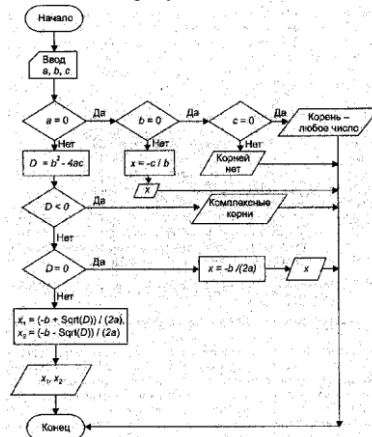
2. Если $D = 0$, то имеется единственный корень (точнее, двукратный корень):

$$x = -\frac{b}{2a}.$$

3. Если $D < 0$, то вещественных корней нет. Блок схема алгоритма приведена на рисунке:



Следует заметить, что приведённый алгоритм предназначен для решения узкого класса задач – квадратных уравнений с «хорошими» коэффициентами. Если допустить, что коэффициенты могут принимать произвольные вещественные значения, есть опасность, что при определённых значениях коэффициента (например, $a = 0$) возникает аварийная ситуация (деление на ноль). Качественный алгоритм и качественная программа должны быть устойчивыми, то есть при любых входных параметрах завершение работы программы должно быть нормальным, хотя, возможно, и сопровождаться предупреждающим сообщением о некорректности входных данных. Свойством устойчивости обладает алгоритм решения квадратного уравнения, приведённый на рисунке:



Разработанный программистом алгоритм должен давать правильный ответ. Проверка алгоритма может оказаться непростым делом. В простых случаях такая проверка может быть выполнена с помощью заполнения *трассировочной таблицы*. Каждый столбец такой таблицы соответствует определённой переменной, а каждая строка – одному шагу алгоритма. Для заполнения таблицы необходимо шаг за шагом проследить выполнение алгоритма, записывая в таблицу текущие значения выбранных для трассировки переменных. Такой метод позволяет выявить логические ошибки, допущенные при составлении или записи алгоритма, и определить, верен ли окончательный ответ. Составим в качестве примера трассировочную таблицу для алгоритма Герона вычисления квадратного корня из числа 2.

i	z
0	1,00000
1	1,50000
2	1,41666
3	1,41421
4	1,41421
5	1,41421

Как видно из таблицы, уже после третьей итерации приближенное значение квадратного корня отличается от точного 1,414213 лишь в шестом знаке после запятой.

Тема 1.4. Язык программирования Qbasic (лекция – дискуссия 10 часов)

История создания языка программирования Qbasic

Язык QBASIC (Beginner's All-purpose Instruction Code) разработан Джоном Кимини и Томасом Куртцем в Дартмутском колледже, США, в середине 1960 г.

QBASIC занимает особое место среди всех языков высокого уровня. С самого начала он задумывался как универсальный язык для начинающих, и средства программирования на QBASIC до сих пор включаются в комплект поставки ПК, - как обязательный элемент технологии.

Вот некоторые достоинства QBASIC (с точки зрения массового пользователя):

- простота синтаксиса;
- простота организации данных и управляющих структур;
- 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).

QBASIC является приложением MS-DOS, которое нормально работает под Windows. Чтобы начать работу с QBASIC, надо запустить программу QBASIC.EXE.

Запускать QBASIC любым удобным способом:

- ◆ Нажатием {Enter} в NC;
- ◆ С помощью меню пользователя в NC;
- ◆ Из папки {языки программирования}, ярлык {QBASIC}.

После запуска QBASIC на экране появится окно. Для снятия заставки и перехода в основное окно редактора нажмите {Esc}. Завершение работы – (Файл-Выход)

Главное окно программной оболочки QBASIC

Главное меню содержит несколько пунктов (Файл, Редактирование, Просмотр и т.п.). Для выбора пункта меню щелкнуть мышью на его название или нажать клавишу {Alt}.

После выбора пункта меню раскрывается нисходящее меню со списком команд данного пункта.

Пункт Файл

- ◆ Новый. Работа с неозаглавленной заготовкой нового файла.
- ◆ Открыть. Открыть существующий файл. В открывшемся окне необходимо найти имя существующего файла. Для перемещения по полям используйте мышшь или клавишу {Tab}.
- ◆ Сохранить. На диске сохраняется текущая версия вашей программы. При первоначальном сохранении файла необходимо присвоить имя создаваемому файлу.
- ◆ Сохранить как... откроется стандартное диалоговое окно, в котором надо указать имя. Этой командой можно создать обновленную версию программы, сохранив старую в неприкосновенности.
- ◆ Печать. Печать открытого файла на принтере.
- ◆ Выход. Завершает работу QBASIC.

Пункт Редактирование

Можно переместить или скопировать выделенный фрагмент текста в другое место того же файла или в другой файл. Для выделения фрагмента необходимо подвести курсор к началу выделяемого участка, нажать {Shift} и, не отпуская ее, «закрашивать» участок.

Пункт Запуск

- ◆ Запуск {Shift+F5}. Запуск программы с первой исполняемой команды.

- ◆ Перезапустить. Подготовка программы к запуску с первой исполняемой команды. Эта команда удаляет все введенные вами данные и высвечивает первую исполняемую команду.
 - ◆ Продолжить {F5}. Работа возобновляется с той команды, на которой программа была остановлена.
- Окно среды QBASIC
 После запуска программы QBASIC, на экране появляется основное окно текстового редактора.
 Основные операции с программой проводятся в окне редактора: набирается, редактируется текст, отлаживается программа и т. п.
 Результаты своей работы программа выдает на выходной экран QBASIC, который “скрыт” под окном редактора. Выходной экран выдается и после завершения программы, - для возврата в окно редактора необходимо нажать любую клавишу.
 F4 – универсальный переключатель между окном редактора и выходным экраном QBASIC.



Алфавит языка Qbasic

Арифметические выражения

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 выполняются 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**.

Встроенные функции языка Qbasic

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

Например. Записать по правилам Бейсика математические выражения

- 1) $\frac{X^2+2X-5.12}{X^2+12.51}$ $(X^2+2*X-5.12)/(X^2+12.51)$
- 2) $\frac{\cos(X) - \sin(X)}{|\cos(X) + \sin(X)|}$ $(\cos(X)-\sin(X))/(ABS(\cos(X)+\sin(X)))$

Упражнения

1. Какие из следующих последовательностей символов являются идентификаторами, а какие нет.

- а) X б) X1 в) X' г) X1X2 д) AB е) ABCD
 ж) SIN з) SIN (X) и) A-1 к) 2A л) MAX15

2. Записать по правилам Бейсика следующие выражения:

- а) $\frac{X^2+Y^2}{1+X^2-Y^2}$ б) $1+X+X^2$ в) $1+|X|+|1+X|$ г) $\frac{A+B}{2}$ C+D
 - 2,5
 д) $\frac{A+B-1.7}{C+D+0.5}$ е) $\frac{1.2-9.8X}{1-Y(54,264-X)}$ ж) $\frac{\sqrt{X^2+Y^2+100}}{(2X+50,2)}$ + 30,2

3. Переписать следующие выражения, записанные по правилам Бейсика, в традиционной математической форме:

- а) $A+B/(C+D) - (A+B)/C+D$ б) $A*B/(C+D) - (C-D)/B*(A+B)$
 в) $1+\text{SQR}(\cos(X+Y)/2)$ г) $2.56 + \text{ABS}(X^2-Y^6)$
 д) $\text{INT}(X*5.234 - A*(X+Y))$

Оператор присваивания

имя переменной = { имя другой переменной
 арифметические выражения
 число


```

    F=Z^2-1
ELSE
    F=1/(1-Z)
ENDIF
PRINT "F=";F
END

```

2 вариант

```

IF Z<0 THEN F=Z-5/Z^2
IF Z<=1 AND Z>=0 THEN F=Z^2-1
IF Z>1 THEN F=1/(1-Z)
PRINT "F=";F
END

```

Для самостоятельного решения:

Составить программу определения кислотности раствора с помощью лакмусовой бумажки. Напомним, что, если бумажка окрашивается в красный цвет - раствор кислотный; синий - раствор щелочной; не меняет цвет - раствор нейтральный.

Множественный выбор (SELECT CASE)

Формат

```

SELECT CASE арифм_выражение или симв_выражение
CASE условие 1
<блок команд 1>
CASE условие 2
<блок команд 2>

```

```

.....
[CASE ELSE
<блок команд n>]
END SELECT

```

Условие оператора CASE можно указать в одном из трех форматов:

- 1) CASE константа 1, константа 2, ...
- 2) CASE IS знак_отношения константа
- 3) CASE константа 1 TO константа 2

Константы в условии должны быть того же типа, что и выражение в SELECT CASE.

Алгоритм множественного выбора заключается в следующем:

- вычисляется значение выражения, записанного в SELECT CASE.
- проверяется, удовлетворяет ли это значение одному из указанных в CASE условий
- если значение удовлетворяет какому-то условию, выполняется блок команд, следующий за данным CASE.

Пример

```

INPUT "Введите значение ",A
SELECT CASE A
CASE 1,5
    PRINT "A равно 1 или 5"
CASE IS >5
    PRINT "A больше 5"
CASE -8 TO 2.5
    PRINT "A не меньше -8, но не больше 2.5"
CASE else
    PRINT "Ни одно условие не выполняется"
END SELECT

```

После выполнения того или иного блока команд управление передается команде, следующей за END SELECT.

Циклы. (с предусловием, с постусловием, с перебором)

Цикл с предусловием

CLS

```

x0 = 1
x1 = 10
dx = 1
a = 1
b = 5
c = -3

```

x = x0

DO WHILE x <= x1 'UNTIL x > x1

```

IF x < 5 THEN y = a
IF x = 5 THEN y = b
IF x > 5 THEN y = c
PRINT "pri x="; x; " y="; y
x = x + dx

```

LOOP

CLS

```

x0 = 1
x1 = 10
dx = 1

```

```

a = 1
b = 5
c = -3

```

x = x0

1 :

IF x > x1 THEN GOTO 2

```

IF x < 5 THEN y = a
IF x = 5 THEN y = b
IF x > 5 THEN y = c
PRINT "pri x="; x; " y="; y
x = x + dx
GOTO 1
2 :

```

Цикл с постусловием

```
CLS
```

```
x0 = 1
x1 = 10
dx = 1
```

```
a = 1
b = 5
c = -3
```

```
x = x0
```

```
DO
```

```

IF x < 5 THEN y = a
IF x = 5 THEN y = b
IF x > 5 THEN y = c
PRINT "pri x="; x; " y="; y
x = x + dx
LOOP UNTIL x > x1 'while x<=x1

```

```
CLS
```

```
x0 = 1
x1 = 10
dx = 1
```

```
a = 1
b = 5
c = -3
```

```
x = x0
```

```
1 :
```

```

IF x < 5 THEN y = a
IF x = 5 THEN y = b
IF x > 5 THEN y = c
PRINT "pri x="; x; " y="; y
x = x + dx

```

```
IF x <= x1 THEN GOTO 1
```

Цикл с перебором

```
CLS
```

```
x0 = 1
x1 = 10
dx = 1
```

```
a = 1
b = 5
c = -3
```

```
FOR x = x0 TO x1 STEP dx
```

```

IF x < 5 THEN y = a
IF x = 5 THEN y = b
IF x > 5 THEN y = c
PRINT "pri x="; x; " y="; y
NEXT x

```

Массивы данных.

Массив (в некоторых [языках программирования](#) также *таблица*, *ряд*, матрица) — набор компонентов (**элементов**), расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по [индексу](#) (индексам). В отличие от [списка](#), массив является структурой с произвольным доступом^[1].

Размерность массива — количество индексов, необходимое для однозначного доступа к элементу массива^{[2][3]}.

Форма или **структура массива** — количество размерностей и размер (протяжённость) массива для каждой размерности^[4], может быть представлен одномерным массивом^[5].

В языке программирования [APL](#) массив является основным типом данных (при этом нуль-мерный массив называется скаляром, одномерный — **вектором**, двумерный — **матрицей**)^[5].

Массив — упорядоченный набор данных, для хранения данных одного типа, идентифицируемых с помощью одного или нескольких *индексов*. В простейшем случае массив имеет постоянную длину и хранит единицы данных одного и того же типа.

Количество используемых индексов массива может быть различным. Массивы с одним индексом называют *одномерными*, с двумя — *двумерными* и т. д. Одномерный массив (колонка, столбец) нестрого соответствует вектору в математике, двумерный — матрице. Чаще всего применяются массивы с одним или двумя индексами, реже — с тремя, ещё большее количество индексов встречается крайне редко.

Одномерный массив

Таблицей или массивом называют набор однотипных данных расположенных в несколько строк одинаковой длины.

Таблицей или массивом называют набор однотипных данных расположенных в несколько строк одинаковой длины.

Бывают одномерные и двумерные массивы.

A = {1,3,-4,100} одномерный массив размерности 4
 1 1 3 4
 A = 3 8 1 5 двумерный массив размерности 3x4
 1 1 -2 2

Таблицей или массивом называют набор
 однотипных данных расположенных в
 несколько строк одинаковой длины.

Бывают одномерные и двумерные массивы.

A = {1,3,-4,100} одномерный массив размерности 4
 1 1 3 4
 A = 3 8 1 5 двумерный массив размерности 3x4
 1 1 -2 2

Одномерным массивом размерности n называется массив содержащий n элементов. Доступ к элементу одномерного массива осуществляется по его номеру. Запись A[i] означает, что элемент стоит на i-том месте. i- это индекс одномерного массива. Одномерным массивом размерности n называется массив содержащий n элементов. Доступ к элементу одномерного массива осуществляется по его номеру. Запись A[i] означает, что элемент стоит на i-том месте. i- это индекс одномерного массива.

A = {1,3,-4,100}

Примеры обработки массивов

CLS

INPUT "Vvedite razmer massiva"; n

DIM a(n)

FOR i = 1 TO n

a(i) = INT(RND(1) * 100 + 1)

NEXT i

PRINT "Massiv A:"

FOR i = 1 TO n

PRINT a(i); " ";

NEXT i

'minimum \ maximum

PRINT

m = a(1): xm = 1

FOR i = 2 TO n

IF a(i) < m THEN m = a(i): xm = i

NEXT i

PRINT "Min element paven "; m; "ego nomer "; xm

' summa\proizvedenie

s = 0: p = 1

FOR i = 1 TO n

s = s + a(i)

p = p * a(i)

NEXT i

PRINT "summa mas ="; s

PRINT "proizv mas ="; p

'sortirovka

FOR k = 1 TO n

FOR i = 2 TO n

IF a(i) > a(i - 1) THEN SWAP a(i), a(i - 1) 'ubivanie

NEXT i

NEXT k

PRINT "Massiv A:"

FOR i = 1 TO n

PRINT a(i); " ";

NEXT i

'kolichestvo chetnix

PRINT

k = 0

FOR i = 1 TO n

IF a(i) MOD 2 = 0 THEN k = k + 1 ' a(i)<0 a(i)=0 a(i)>10

NEXT i

PRINT "kol-vo chetnix ="; k

' zamena

FOR i = 1 TO n

IF a(i) > 50 THEN a(i) = 50

NEXT i

PRINT

PRINT "Massiv A:"

FOR i = 1 TO n

PRINT a(i); " ";

NEXT i

Двумерный массив

Двумерный массив - это [одномерный массив](#), элементами которого являются одномерные массивы. Другими словами, это набор однотипных данных, имеющий общее имя, доступ к элементам которого осуществляется по двум индексам. Наглядно двумерный массив удобно представлять в виде таблицы, в которой n строк и m столбцов, а под ячейкой таблицы, стоящей в i -й строке и j -м столбце понимают некоторый элемент массива $a[i][j]$.

Действительно, если разобраться с тем, что такое $a[i]$ при фиксированном значении i , то увидим, что это одномерный массив, состоящий из m элементов, к которым можно обращаться по индексу: $a[i][1]$, $a[i][2]$, ..., $a[i][m]$. Схематически это вся i -я строка строки таблицы. Аналогично, если мы рассмотрим одномерный массив строк, то сможем заметить, что это так же двумерный массив, где каждый отдельный элемент - это символ типа char, а $a[i]$ - это одномерный массив, представляющий отдельную строку исходного одномерного массива строк. Исходя из идеи определения двумерного массива можно определить рекуррентное понятие многомерного массива:

n-мерный массив - это одномерный массив, элементами которого являются $(n-1)$ -мерные массивы.

Несложно догадаться, что 3-мерный массив визуально можно представить в виде куба с ячейками (похоже на кубик Рубика), где каждый элемент имеет вид $a[i][j][k]$. А вот с большими размерностями возникают сложности с визуальным представлением, но математическая модель ясна.

По-другому двумерный массив также называют **матрицей**, а в том случае, когда $n=m$ (число строк равно числу столбцов) матрицу называют квадратной. В матрицах можно хранить любые табличные данные: содержание игрового поля (шашки, шахматы, Lines и т.д.), лабиринты, таблицу смежности графа, коэффициенты системы линейных уравнений и т.д. Матрицы часто используют для решения олимпиадных и математических задач.

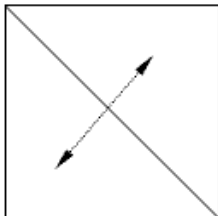
В задачах табличные данные часто определяются во входном файле следующим образом: сначала в первой строке указываются значения n и m через пробел, а далее идут n строк по m элементов в каждой, также друг от друга отделенные пробелом и входной файл может иметь, например, следующее содержание, понятно отражающее содержимое матрицы при обычном просмотре:

```
3 5
7 8 2 3 1
5 3 2 6 3
9 3 5 2 0
```

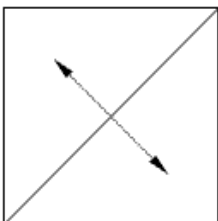
В приведенном примере определена матрица, состоящая из трех строк и пяти столбцов. Рассмотрим пример чтения этих данных в матрицу и вывода матрицы в файл. Для этого удобно использовать двойной цикл, где внешний цикл по i будет пробегать по всем строкам, а внутренний цикл по j будет для текущей строки i перебирать все ее элементы. Алгоритмическая реализация этого процесса может выглядеть следующим образом:

```
int a[1..100][1..100];
//Чтение данных двумерного массива из файла
read(n,m);
for i=1..n{
  for j=1..m{
    read(a[i][j]);
  }
}
//Вывод матрицы в файл
for i=1..n{
  for j=1..m{
    write(a[i][j], ' ');
  }
  writeln; //новая строка
}
```

В качестве примера рассмотрим задачу о транспонировании квадратной матрицы относительно главной и побочной диагонали, где необходимо симметричным образом поменять элементы двумерного массива относительно одной из диагонали. Алгоритмическое решение данной задачи может быть представлено следующим образом:



```
for i=2..n{
  for j=1..i-1{
    a[i][j] <---> a[j][i];
  }
}
```



```
for i=1..n-1{
  for j=1..n-i{
    a[i][j] <---> a[n-j+1][n-i+1];
  }
}
```

Примеры обработки массивов.

```
CLS
'INPUT "vvvwdite razmer massiva n x n"; n
```

```
n = 5
```

```
DIM a(n, n), s(n), p(n)
```

```
'zapolnenie
FOR i = 1 TO n
FOR j = 1 TO n
a(i, j) = INT(RND(1) * 100 + 1)
NEXT j
NEXT i
```

```
'pechat
PRINT "massiv a:"
```

```

FOR i = 1 TO n
FOR j = 1 TO n
LOCATE , j * 3: PRINT a(i, j);
NEXT j
PRINT
NEXT i

```

'summa strok \stolbsov

```

FOR i = 1 TO n
FOR j = 1 TO n
s(i) = s(i) + a(i, j) ' s(j)=s(j)+a(i,j)
NEXT j
NEXT i
PRINT

```

```

FOR i = 1 TO n
PRINT "summa stroki "; i; " ravna"; s(i)
NEXT i
PRINT

```

'proizvedenie stroki\stolbsa

```

FOR i = 1 TO n
p(i) = 1
NEXT i

```

```

FOR i = 1 TO n
FOR j = 1 TO n
p(i) = p(i) * a(i, j) ' p(j)=p(j)*a(i,j)
NEXT j
NEXT i

```

PRINT

```

FOR i = 1 TO n
PRINT "proizvedenie stroki "; i; " ravno"; p(i)
NEXT i
PRINT

```

' minimum\maksimum

m = a(1, 1): xm = 1: ym = 1

```

FOR i = 1 TO n
FOR j = 1 TO n
IF a(i, j) < m THEN m = a(i, j): xm = i: ym = j ' a(i,j)>m
NEXT j
NEXT i

```

PRINT "min element raven "; m; "stroka "; xm; "stolbes"; ym

'dioganali

```

sd = 0
FOR i = 1 TO n
sd = sd + a(i, i)
NEXT i

```

PRINT "summa el glavnoy diogonalny ravna "; sd

pd = 1

```

FOR i = 1 TO n
pd = pd * a(i, n - i + 1)
NEXT i

```

PRINT "proizvedenie el pobochnoi dioganaly "; pd

'sortirovka strok\stolbcov

```

FOR k = 1 TO n
FOR i = 2 TO n
FOR j = 1 TO n
IF a(i, j) > a(i - 1, j) THEN SWAP a(i, j), a(i - 1, j) ' a(i, j) < a(i - 1, j)
NEXT j
NEXT i
NEXT k

```

PRINT

PRINT "massiv a:"

```

FOR i = 1 TO n
FOR j = 1 TO n
LOCATE , j * 3: PRINT a(i, j);
NEXT j

```

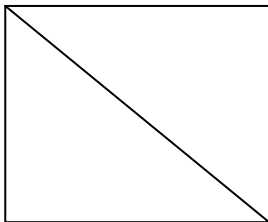
```
PRINT  
NEXT i
```

'oblast' nige pobochnoi diogonaly

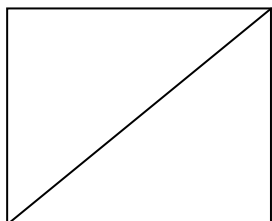
```
FOR i = 1 TO n  
FOR j = 1 TO n  
IF i >= n - j + 1 THEN a(i, j) = 0  
NEXT j  
NEXT i
```

```
PRINT
```

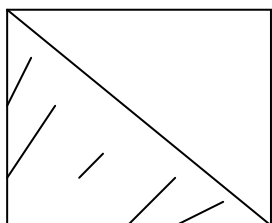
```
PRINT "massiv a:"  
FOR i = 1 TO n  
FOR j = 1 TO n  
LOCATE , j * 3: PRINT a(i, j);  
NEXT j  
PRINT  
NEXT i
```



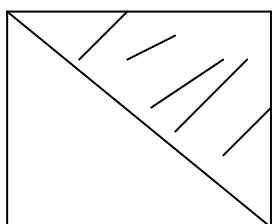
$$i=j$$



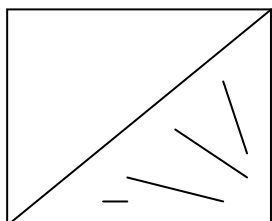
$$i=n-j+1$$



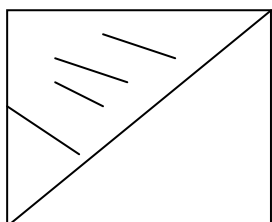
$$i \geq j$$



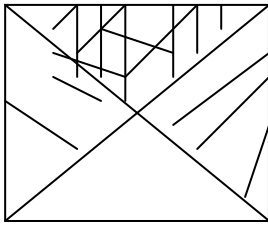
$$i \leq j$$



$$i \geq n-j+1$$

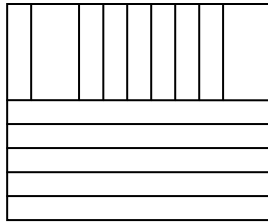


$$i \leq n-j+1$$



I<=j

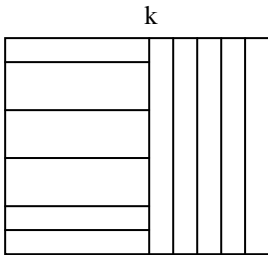
and
I<=n-j+1



k

I<=k

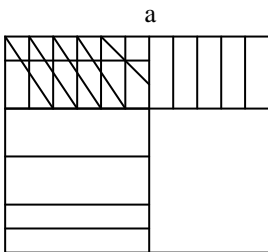
I>=k



k

j<=k

j>=k



a

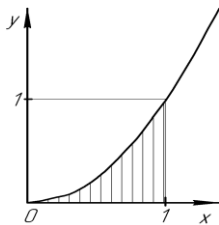
b

i<=b

J<=a
and

Решение практических задач.

Нахождение площади криволинейной трапеции методом прямоугольников.



n = 1000000

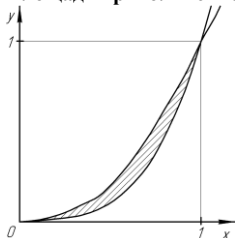
```
FOR i = 1 TO n
x = RND(1)
y = RND(1)
IF y <= x ^ 2 THEN k = k + 1
NEXT i
```

```
s = 1 * k / n
PRINT "ploshad = "; s
```

Результаты работы программы

При n=100 s=0.33
При n=1000 s=0.319
При n=10000 s=0.3305
При n=100000 s=0.33346
При n=1000000 s=0.333744

Площадь криволинейной трапеции



$$y=x^2$$

$$s = \int_0^1 x^2 dx = \frac{x^3}{3} \Big|_0^1 = \frac{1}{3} - \frac{0}{3} = 0,3(3)$$

$$y \geq x^2$$

$$y=x^2$$

$$y=x^3$$

$$S=S_1- S_2$$

$$S_1 = \int_0^1 x^2 dx = \frac{x^3}{3} \Big|_0^1 = \frac{1}{3} - \frac{0}{3} = 0,3(3)$$

$$S_2 = \int_0^1 x^3 dx = \frac{x^4}{4} \Big|_0^1 = \frac{1}{4} - \frac{0}{4} = 0,25$$

$$S = 0,3(3) - 0,25 = 0,08(3)$$

```

n = 1000000
FOR i = 1 TO n
x = RND(1)
y = RND(1)
IF y <= x ^ 2 AND y >= x ^ 3 THEN k = k + 1
NEXT i

```

```

s = 1 * k / n
PRINT "ploshad = "; s

```

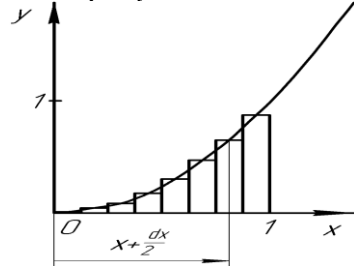
Результаты работы программы

```

при n=100 s = .09
при n=1000 s =.089
при n=10000 s = .0848
при n=100000 s = .08397
при n=1000000 s = .083558

```

Метод прямоугольников



$$S_i = f(x + dx/2) * dx$$

n = 1000000

```

x1 = 0
x2 = 1

```

```

dx = (x2 - x1) / n
s = 0

```

```

FOR x = x1 TO x2 STEP dx
s = s + (x + dx / 2) ^ 2 * dx
NEXT x

```

```

PRINT "ploshad = "; s

```

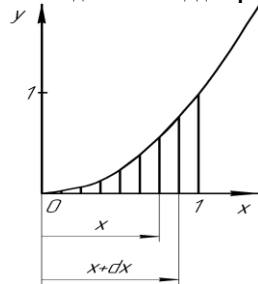
Результаты работы программы

```

при n=100 s = .3434249
при n=1000 s =.3343292
при n=10000 s =.3333427
при n=100000 s = .3328973
при n=1000000 s =.3289963

```

Нахождение площади криволинейной трапеции методом трапеций.



$$S_i = \frac{f(x) + f(x + dx)}{2} * dx$$

n = 100000

```

x1 = 0
x2 = 1

```

```

dx = (x2 - x1) / n
s = 0

```

```

FOR x = x1 TO x2 - dx STEP dx
s = s + ((x ^ 2 + (x + dx) ^ 2) / 2) * dx
NEXT x

```

```

PRINT "ploshad = "; s

```

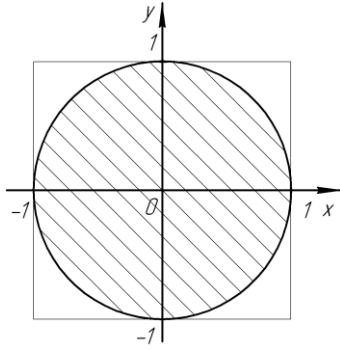
Результаты работы программы

```

при n=100 s = .3333285
при n=1000 s =.3332427
при n=10000 s =.3328873
при n=100000 s = .3328973
при n=1000000 s =.3289963

```


Нахождение числа π методом монте-Карло.



$$\text{Уравнение окружности } x^2 + y^2 = R^2$$

Условие того, что точка (x, y) лежит внутри окружности $x^2 + y^2 \leq R^2$

```
n = 10000000
s = 0
```

```
FOR i = 1 TO n
```

```
x = RND(1) * 2 - 1
y = RND(1) * 2 - 1
IF x ^ 2 + y ^ 2 <= 1 THEN s = s + 1
NEXT i
```

```
pi = 4 * s / n
PRINT "pi="; pi
```

Результаты работы программы

```
При n=100 pi=3.28
При n=1000 pi=3.204
При n=10000 pi=3.16
При n=100000 pi=3.13
При n=1000000 pi=3.1426
При n=2000000 pi=3.142952
При n=10000000 pi=3.141768
```

Подпрограммы

Процедуры

Подпрограммой называется участок программы, оформленный определенным образом, к которому можно обращаться из разных точек программы любое число раз. При этом подпрограмма может решать каждый раз одну и ту же задачу с разными значениями исходных данных. Использование подпрограмм позволяет существенно уменьшить объем рабочей программы. Деление программы на подпрограммы делает программу более понятной.

Обращение к подпрограмме осуществляется с помощью команды

GOSUB номер строки или метка

По этой команде управление передается подпрограмме - блоку команд, первая из которых помечена номером строки или меткой. Последней выполняемой командой подпрограммы является **RETURN**. По этой команде управление передается в основную программу на оператор, стоящий за оператором **GOSUB**.

Если нарушено согласование операторов **RETURN** и **GOSUB**, то выводится сообщение об ошибке "RETURN without GOSUB".

Подпрограммы располагаются в конце основной программы и отделяются от основной программы оператором **END**.

Пример 1. Написать подпрограмму вывода подчеркивания информации. Длина черты переменная.

```
N=5
GOSUB A 'вызов подпрограммы для вывода знака подчеркивания 5 раз
N=30
GOSUB A 'вызов подпрограммы для вывода знака подчеркивания 30 раз
END
A: FOR I=1 TO N 'подпрограмма вывода знака подчеркивания
  PRINT " _";
NEXT I
PRINT
```

```
RETURN 'оператор возврата в основную программу
```

Пример 2. Вычислить функцию: $C = m!n!/(m-n)!$

```
1 INPUT "Введите m>n";m,n
IF M<=N THEN 1 'проверка входных данных
R=M
GOSUB A 'вызов подпрограммы для вычисления m!
R1=F : R=N
GOSUB A 'вызов подпрограммы для вычисления n!
R2=F : R=M-N
GOSUB A 'вызов подпрограммы для вычисления (m-n)!
C=(R1+R2)/F : PRINT "C=";C
END
A: F=1 'подпрограмма вычисления факториала
FOR I=2 TO R
  F=F*I
NEXT I
RETURN 'оператор возврата в основную программу
```

Для самостоятельного решения

Составить программу расчета заработной платы по формуле: оклад * количество дней * 1.4-подходящий налог. Расчет заработной платы оформить в подпрограмме.

Функции.

Помимо стандартных числовых функций (sin, cos и т.д.) или символьных функций пользователь может определить и свои собственные. Иногда при решении задач возникает необходимость вычисления одного и того же выражения при различных значениях величин, входящих в это выражение. Вот тогда и применяются функции, определяемые пользователем.

Формат:

DEF FN <имя функции>(<список формальных параметров>)=<выражение>

Обращение к функции пользователя осуществляется

FN <имя функции>(<список фактических параметров>)

Имя определяемой функции должно быть уникально: оно не может повторять имя другой процедуры, переменной или функции.

Формальные параметры резервируют место в памяти для фактических параметров, которые должны быть определены к моменту выполнения функции. Параметры могут быть любого типа. Примеры определения функции:

```
DEF FNKT(X)=COS(X)/SIN(X)
```

```
DEF FNP(A,B)=A/B*100
```

X,A,B – формальные параметры

Между формальными и фактическими параметрами и их типами существует однозначное позиционное соответствие слева направо. Примеры обращения:

```
FNKT(X)
```

```
FNP(C,D)
```

X,C,D - фактические параметры

Имена формальных и фактических параметров могут не совпадать.

С помощью оператора DEF FN можно описать более сложную функцию, занимающую несколько строк

Формат:

DEF FN <имя функции>(<список аргументов>)

<блок команд>

FN<имя функции>=<выражение>

END DEF

В строке FN<имя функции>=записывается выражение, значение которого функция возвращает.

Пример 1. Отрезки заданы координатами концов. Сравнить эти отрезки.

```
DEF FNL(x1,x2,y1,y2)=SQR((x1-x2)^2+(y1-y2)^2) ' функция нахождения длины отрезка
```

```
INPUT "введите координаты первого отрезка",x1,x2,y1,y2
```

```
L1=FNL(x1,x2,y1,y2)
```

```
INPUT "введите координаты второго отрезка",x1,x2,y1,y2
```

```
L2=FNL(x1,x2,y1,y2)
```

```
IF L1>L2 THEN
```

```
  ? "длина первого отрезка больше"
```

```
ELSEIF L2>L1 THEN
```

```
  ? "длина второго отрезка больше"
```

```
ELSE
```

```
  ? "отрезки равны"
```

```
END IF
```

Пример 2. Вычислить сложное выражение, используя определяемую функцию для вычисления повторяющейся в нем части.

```
K=|(1+m^2)/cos(1-m)|-5*sqrt((c+d^2)/cos(c-d)|+tg|(x+y^2)/cos(x-y)|
```

```
DEF FNT(a,b)=abs((a+b^2)/cos(a-b))
```

```
INPUT "Введите 5 значений",m,c,d,x,y
```

```
K=FNT(1,m)-5*sqrt(FNT(c,d))+tan(FNT(x,y))
```

```
? "K=";K
```

Пример 3. Вычислить сумму N членов ряда. Факториал вычислять с помощью функции пользователя.

```
S=1!/4!+4!/5!+9!/6!+16!/7!+...+(n^2)!/(n+3)!
```

```
DEF FNFACT(M)
```

```
  P=1
```

```
  FOR I=1 TO M
```

```
    P=P*I
```

```
  NEXT I
```

```
  FNFACT=P
```

```
END DEF
```

```
INPUT "Введите n";N
```

```
S=0
```

```
FOR J=1 TO N
```

```
  S=S+FNFACT(J^2)/FNFACT(J+3)
```

```
NEXT J
```

```
? "S=";S
```

Задание:

1. Вычислить сложное выражение, используя определяемую функцию для вычисления повторяющейся в нем части.

$$C = \sqrt{(d + |e|) / \operatorname{tg}(de) + (x + |y|) / \operatorname{tg}(xy) - (z + |t|) / \operatorname{tg}(zt)}$$

2. Вычислить функцию: $C = m! / (m-n)!$. Для вычисления факториала использовать определяемую функцию.

Практическое занятие с использованием подпрограмм и функций пользователя

1 вариант

1. Дан массив с оценками учащихся. Найти средний балл учащихся. Подсчитать количество 5 и 4. Заполнение массива, подсчет среднего балла и подсчет количества оценок оформить отдельными подпрограммами.

2. Задана матрица W(5,6). Отрицательные элементы матрицы заменить на 0. измененный массив вывести на экран. Обработку элементов массива оформить в подпрограмме.

2 вариант

1. Дан одномерный массив со значениями веса кур на птицефабрике. Найти средний вес кур. Найти вес самой тяжелой курицы. Заполнение массива, подсчет среднего веса и нахождение веса самой тяжелой курицы оформить отдельными подпрограммами.

2. Найти произведение элементов массива R(7,5) в строках с четными номерами. Обработку элементов массива оформить в подпрограмме.

Графика в Qbasic Режимы экрана



В следующей таблице дана обобщенная информация по режимам экрана:

MDPA, CGA, Hercules, Olivetti, EGA, VGA или MCGA Адаптеры

SCREEN 0: Текстовый режим

- 40 x 25, 40 x 43, 40 x 50, 80 x 25, 80 x 43 или 80 x 50 формат текста
8 x 8 формат символа (8 x 14, 9 x 14 или 9 x 16 с EGA или VGA)
- 16 цветов назначаются любому из 16 атрибутов (с CGA или EGA)
- 64 цвета назначаются любому из 16 атрибутов (с EGA или VGA)
- В зависимости от разрешения текста и адаптера, 8 страниц видеопамати (0-7), 4 страницы (0-3), 2 страницы (0-1) или 1 страница (0)

CGA, EGA, VGA или MCGA Адаптеры

SCREEN 1: 320 x 200 графический режим

- 40 x 25 формат текста, 8 x 8 формат символа
- 16 фоновых цветов и один из двух наборов 3 цветов переднего плана, присвоенных с использованием оператора COLOR с CGA
- 16 цветов назначаются 4 атрибутам с EGA или VGA
- 1 страница видеопамати (0)

SCREEN 2: 640 x 200 графический режим

- 80 x 25 формат текста, 8 x 8 формат символа
- 16 цветов назначаются 2 атрибутам с EGA или VGA
- 1 страница видеопамати (0)

Hercules, Olivetti или AT&T Adapters

SCREEN 3: Нужен адаптер Hercules, только монохромный монитор

- 720 x 348 графический
- 80 x 25 формат текста, 9 x 14 формат символа
- Обычно 2 страницы видеопамати (0-1); 1 страница (0), если установлен второй цветной видеоадаптер
- Оператор PALETTE не поддерживается
- Вызовите Hercules драйвер MSHERC.COM перед использованием режима экрана 3

SCREEN 4:

- Поддерживает персональные компьютеры фирмы Olivetti моделей M24, M240, M28, M280, M380, M380/C и M380/T, а также персональные компьютеры фирмы AT&T серии 6300
- 640 x 400 графический режим
- 80 x 25 формат текста, 8 x 16 формат символа
- 1 из 16 цветов присвоен цвету переднего плана (выбранный оператором COLOR); фон - неизменно черный
- 1 страница видеопамати (0)
- Оператор PALETTE не поддерживается

EGA или VGA Адаптеры

SCREEN 7: 320 x 200 графический режим

- 40 x 25 формат текста, 8 x 8 формат символа
- Присвоение 16 цветов любому из 16 атрибутов
- При 64К памяти EGA адаптера - 2 страницы видеопамати (0-1); в других случаях - 8 страниц (0-7)

SCREEN 8: 640 x 200 графический режим

- 80 x 25 формат текста, 8 x 8 формат символа
- Присвоение 16 цветов любому из 16 атрибутов
- При 64К памяти EGA адаптера - 1 страница видеопамати (0); в других случаях, 4 страницы (0-3)

SCREEN 9: 640 x 350 графический режим

- 80 x 25 или 80 x 43 формат текста, 8 x 14 или 8 x 8 формат символа
- 16 цветов назначаются 4 атрибутам (память адаптера 64К) или 64 цвета назначаются 16 атрибутам (память адаптера более 64К)
- При 64К памяти EGA адаптера - 1 страница видеопамати (0); в других случаях, 2 страницы (0-1)

EGA или VGA Адаптеры, Только Монохромные Мониторы

SCREEN 10: 640 x 350 графический режим, только монохромные мониторы

- 80 x 25 или 80 x 43 формат текста, 8 x 14 или 8 x 8 формат символа
- Вплоть до 9 псевдоцветов назначаются 4 атрибутам
- 2 страницы видеопамати (0-1), нужен адаптер с 256К памяти

VGA или MCGA Адаптеры

Screen 11 (VGA или MCGA)

- 640 x 480 графический режим

- 80 x 30 или 80 x 60 формат текста, 8 x 16 или 8 x 8 формат символа
 - Присвоение вплоть до 256K цветов 2 атрибутам
 - 1 страница видеопамати (0)
- Screen 12 (VGA)
- 640 x 480 графический режим
 - 80 x 30 или 80 x 60 формат текста, 8 x 16 или 8 x 8 формат символа
 - Присвоение вплоть до 256K цветов 16 атрибутам
 - 1 страница видеопамати (0)
- Screen 13 (VGA или MCGA)
- 320 x 200 графический режим
 - 40 x 25 формат текста, 8 x 8 формат символа
 - Присвоение вплоть до 256K цветов 256 атрибутам
 - 1 страница видеопамати (0)

Цвета

Устанавливает цвета экрана.

COLOR [основной%] [, [фоновый%] [, граница%]] Режим экрана 0 (только текст)
 COLOR [фоновый%] [, палитра%] Режим экрана 1
 COLOR [основной%] Режимы экрана 4, 12, 13
 COLOR [основной%] [, фоновый%] Режимы экрана 7-10

- **основной%** Число, устанавливающее цвет переднего плана экрана.
 основной& В режиме экрана 0 основной% - это атрибут цвета, устанавливающий цвет текста. В других режимах экрана основной% - это атрибут цвета или 4-битовое значение (только в режиме экрана 4), устанавливающие цвет текста и линий.
- **фоновый%** Число, устанавливающее фоновый цвет экрана. В режиме экрана 0, фоновый% - это атрибут цвета. В режиме экрана 1, фоновый% - это 4-битовое значение. В режимах экрана 7-10, фоновый& - это значение цвета.
- **граница%** Атрибут цвета, устанавливающий цвет границы экрана.
- **палитра%** Число (0 или 1), указывающее какую из двух групп атрибутов цвета нужно использовать:

палитра%	Атрибут 1	Атрибут 2	Атрибут 3
0	Зеленый	Красный	Коричневый
1	Бирюзовый	Малиновый	Ярко-белый

- Допустимые атрибуты и значения цвета зависят от Вашего графического адаптера и режима экрана, установленного последним оператором SCREEN.
- Если в Вашей системе есть EGA, VGA или MCGA адаптер, используйте оператор PALETTE для изменения назначений цвета для атрибутов цвета.

Пример:

'Для выполнения этого примера нужен цветной графический адаптер.

```
SCREEN 7
FOR i% = 0 TO 15
  COLOR i%
  PRINT i%
```

```
NEXT i%
  Цветной видеоадаптер      Монохромный видеоадаптер
```

Атрибут цвета	Значение цвета по умолчанию(a)	Выводимый цвет	Значение цвета по умолчанию	Выводимый цвет
---------------	--------------------------------	----------------	-----------------------------	----------------

SCREEN Режимы 0, 7, 8, 9(b), 12 13				
0	0	Черный	0(c)	Выкл
1	1	Голубой		Подчеркнутый(d)
2	2	Зеленый	1(c)	Вкл(d)
3	3	Бирюзовый	1(c)	Вкл(d)
4	4	Красный	1(c)	Вкл(d)
5	5	Малиновый	1(c)	Вкл(d)
6	6	Коричневый	1(c)	Вкл(d)
7	7	Белый	1(c)	Вкл(d)
8	8	Серый	0(c)	Выкл
9	9	Светло-голубой		Яркий
				Подчеркнутый
10	10	Светло-зеленый	2(c)	Яркий
11	11	Светло-бирюзовый	2(c)	Яркий
12	12	Светло-красный	2(c)	Яркий
13	13	Светло-малиновый	2(c)	Яркий
14	14	Желтый	2(c)	Яркий
15	15	Ярко-белый	0(c)	Выкл

SCREEN Режимы 1 и 9(e)

0	0	Черный	0	Выкл
1	11	Светло-бирюзовый	2	Яркий
2	13	Светло-малиновый	2	Яркий
3	15	Ярко-белый	0	Выкл белый

SCREEN Режимы 2 и 11				
0	0	Черный	0	Выкл
1	15	Ярко-белый	0	Выкл белый

- (a) Номера цвета для EGA, VGA и MCGA используют те значения выводимых цветов, которые создают визуально-одинаковые цвета.
- (b) Для VGA или EGA с видео-памятью > 64К.
- (c) Только для режима 0.
- (d) Выкл., когда используется как фон.
- (e) EGA с видеопамтью <= 64К.

Устанавливает режим и другие характеристики экрана.

SCREEN режим% [, [перекл_цвета%] [, [актив_стр%] [, [видим_стр%]]]

- режим% Устанавливает режим экрана. См. □ Режимы Экрана □ .
- перекл_цвета% Значение (0 или 1), которое переключает между цветным и монохромным изображением (только в режимах 0 и 1):

Режим	Значение	Действие
0	0	Отключает цвет
0	Не ноль	Включает цвет
1	0	Включает цвет
1	Не ноль	Отключает цвет

- актив_стр% Страница экрана, в которую записывается вывод текста или графики.
- видим_стр% Страница экрана, отображаемая на экране в данный момент.

Пример:

'Для выполнения этого примера нужен цветной графический адаптер.

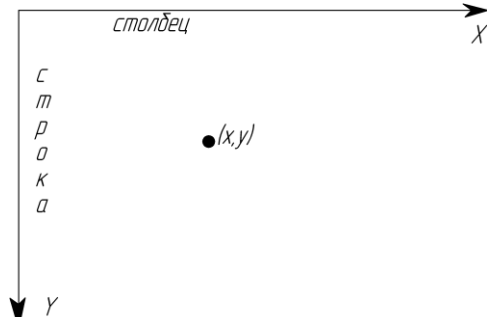
SCREEN 1 '320 x 200 graphics

LINE (110, 70)-(190, 120), , B

LINE (0, 0)-(320, 200), 3, , &HFF00

Точка, линия, квадрат, окружность, дуга, заливка цветом

Точка



Рисует заданную точку на экране.

PRESET [STEP] (x!,y!) [,color%]

PSET [STEP] (x!,y!) [,color%]

- STEP Указывает, что x! и y! заданы относительно текущего графического положения курсора.
- (x!,y!) Координаты экрана устанавливаемой точки раstra.
- color% Атрибут цвета, устанавливаемый для точки раstra. Если color% опущен, PRESET использует текущий цвет фона, а PSET использует текущий цвет переднего плана.
- Доступные атрибуты цвета зависят от Вашего графического видеоадаптера и режима экрана. Значения координат зависят от графического видеоадаптера, режима экрана и последних установок в операторах VIEW и WINDOW.

Пример:

'Для выполнения этого примера нужен графический видеоадаптер.

SCREEN 1

FOR i% = 0 TO 320

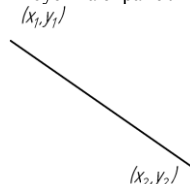
 PSET (i%, 100)

 FOR delay% = 1 TO 100: NEXT delay%

 PRESET (i%, 100)

NEXT i%

Рисует на экране **линию или прямоугольник**.



LINE [[STEP](x1!,y1!)-[STEP](x2!,y2!) [, [цвет%] [, [B | BF] [, [стиль%]]]]

- STEP Указывает, что координаты задаются относительно текущего графического положения курсора.
- (x1!,y1!), (x2!,y2!) Координаты начала и конца линии на экране.
- цвет% Атрибут цвета, устанавливающий цвет линии или прямоугольника. Допустимые атрибуты цвета зависят от Вашего графического адаптера и режима экрана, установленного последним оператором SCREEN.
- B Рисует прямоугольник вместо линии.
- BF Рисует заполненный прямоугольник.
- стиль% 16-битовое значение, биты которого устанавливают будут ли рисоваться точки растра. Используется для изображения прерывистых и пунктирных линий.

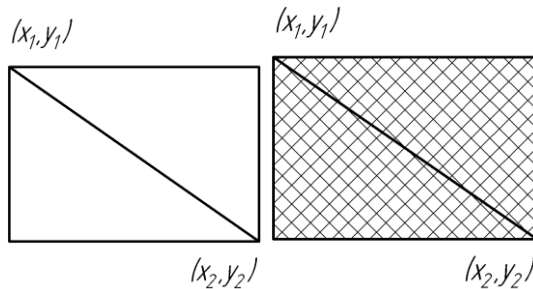
Пример:

Для выполнения этого примера нужен цветной графический адаптер.

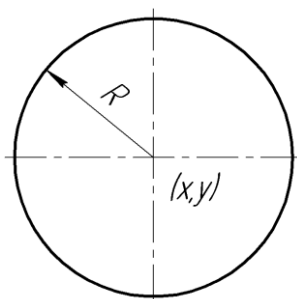
SCREEN 1

LINE (110, 70)-(190, 120), B

LINE (0, 0)-(320, 200), 3, , &HFF00



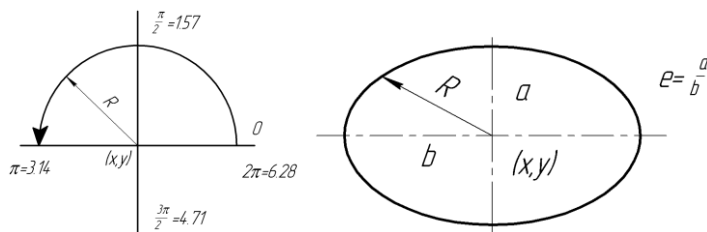
Рисует на экране окружности и эллипсы.



CIRCLE [STEP] (x!,y!),радиус![, [цвет%] [, [старт!] [, [конец!] [, [аспект!]]]]

- STEP Указывает, что координаты задаются по отношению к текущей графической позиции курсора.
- (x!,y!) Координаты центра окружности или эллипса.
- радиус! Радиус окружности или эллипса в единицах текущей системы координат, определенной последними операторами SCREEN, VIEW и WINDOW.
- цвет% Атрибут цвета, устанавливающий цвет окружности. Допустимые атрибуты цвета зависят от Вашего графического адаптера и режима экрана, установленного последним оператором SCREEN.
- старт! Начальный угол дуги в радианах.
- конец! Конечный угол дуги в радианах.
- аспект! Отношение длины оси Y к длине оси X, используемое при изображении эллипсов.

Для перевода градусов в радианы умножьте значение в градусах на $(\pi / 180)$.



Пример:

Для выполнения этого примера нужен графический видеоадаптер.

SCREEN 2

CIRCLE (320, 100), 200

CIRCLE STEP (0,0), 100

Закраска области

Заполняет графическую область указанным цветом или образцом.

PAINT [STEP] (x!,y!)[, [цвет% | образец\$] [, [цвет_границ%] [, [фон\$]]]

- STEP Указывает, что координаты задаются относительно текущей графической позиции курсора.
- (x!,y!) Координаты экрана начала зарисовки.
- color% Атрибут цвета, устанавливающий заполняющий цвет.
- образец\$ Образец заполнения шириной 8 бит и длиной до 64

байт, определяемый следующим образом:
образец\$ = CHR\$(arg1) + CHR\$(arg2) + ... + CHR\$(argn%)
Аргументы CHR\$ - числа в пределах от 0 до 255.
Каждый CHR\$(argn%) определяет 1-байт, участок в 8 пикселей образца, основанного на бинарной форме числа.

- цвет_границ% Атрибут цвета, указывающий цвет границы заполняемой области. PAINТ прекращает заполнение области, когда он наталкивается на границу указанного цвета.
 - фон\$ 1 байт, фон участка в 8 пикселей образца. Указание фона участка образца позволяет Вам зарисовывать уже зарисованную область.
- Допустимые атрибуты цвета зависят от Вашего графического адаптера и режима экрана, установленного последним оператором SCREEN.

Пример:

'Для выполнения этого примера нужен цветной графический адаптер.
SCREEN 1
CIRCLE (106, 100), 75, 1
LINE (138, 35)-(288, 165), 1, B
PAINT (160, 100), 2, 1

Очистка экрана

CLS [{0 | 1 | 2}]

CLS Очищает либо текстовую, либо графическую области просмотра.

Если была установлена графическая область просмотра (с помощью VIEW), то очищает только графическую область просмотра. В противном случае, очищает текстовую область просмотра или весь экран.

CLS 0 Полностью очищает экран, как текст, так и графику.

CLS 1 Очищает графическую область просмотра или весь экран, если графическая область просмотра не была установлена.

CLS 2 Очищает текстовую область просмотра.

Построение графиков функции

$R(t)=(1+\sin(t))(1+0.9\cos(8t))(1+0.1\cos(24t))$

Кардиоида

$R=a(1+\cos(t))$

$(x^2+y^2)^2-2ax(x^2+y^2)-a^2y^2=0$

Четырёхлепестковая роза

$(x^2+y^2)^3=4a^2x^2y^2$

Окружность

$X^2+Y^2=R^2$

$y^2=R^2-x^2$

$y = \pm\sqrt{R^2-x^2}$

$R \sin(t)$
 $y = \sqrt{R^2-x^2}$

$y = -\sqrt{R^2-x^2}$

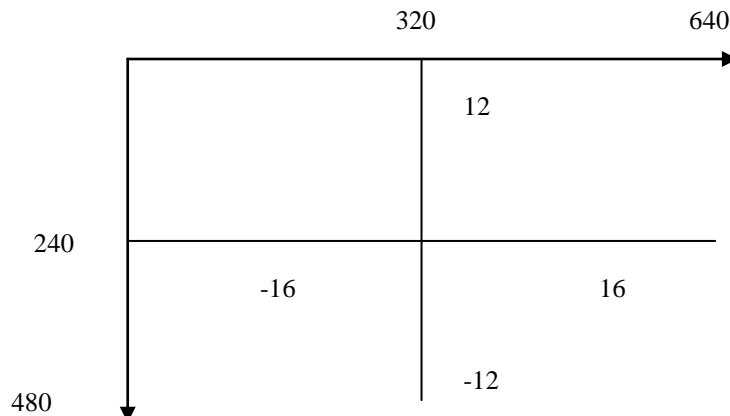
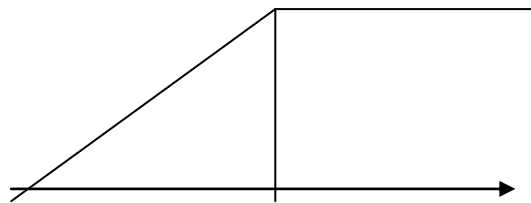
область допустимых значений

$R^2-x^2 \geq R \cos(t)$

$R^2 \geq x^2$

$|x| \leq R$

$x \in (-R, R)$



SCREEN 12 '640x480

'центр (320,240)

LINE (0, 240)-(640, 240), 15

LINE (320, 0)-(320, 480), 15

FOR x = 0 TO 640 STEP 20

LINE (x, 235)-(x, 245), 15

```

NEXT x

FOR y = 0 TO 480 STEP 20
LINE (315, y)-(325, y), 15
NEXT y

' x^2
FOR x = -16 TO 16 STEP .005
PSET (320 + x * 20, 240 - (x ^ 2) * 20), 1
NEXT x

' sin x
FOR x = -16 TO 16 STEP .005
PSET (320 + x * 20, 240 - SIN(x) * 20), 2
NEXT x

' cos x
FOR x = -16 TO 16 STEP .005
PSET (320 + x * 20, 240 - 3 * COS(x) * 20), 3
NEXT x

'окружность

FOR x = -5 TO 5 STEP .005
PSET (320 + x * 20, 240 - SQR(25 - x ^ 2) * 20), 4
PSET (320 + x * 20, 240 + SQR(25 - x ^ 2) * 20), 4
NEXT x

'гипербола
FOR x = .05 TO 16 STEP .001
PSET (320 + x * 20, 240 + 3 * (1 / x) * 20), 5
PSET (320 - x * 20, 240 - 3 * (1 / x) * 20), 5
NEXT x

SCREEN 12 '640x480
CLS

'центр (320,240)

LINE (0, 240)-(640, 240), 15
LINE (320, 0)-(320, 480), 15

FOR x = 0 TO 640 STEP 20
LINE (x, 235)-(x, 245), 15
NEXT x

FOR y = 0 TO 480 STEP 20
LINE (315, y)-(325, y), 15
NEXT y

' четырехлепестковая роза
'FOR x = -10 TO 10 STEP .05
'FOR y = -10 TO 10 STEP .05
'
'IF (x ^ 2 + y ^ 2) ^ 3 < (4 * (10) ^ 2 * x ^ 2 * y ^ 2) THEN PSET (320 + x * 20, 240 - y * 20), 1
'
'NEXT y
'NEXT x

' кардиоида
'FOR x = -10 TO 10 STEP .05
'FOR y = -10 TO 10 STEP .05
'
'IF (x ^ 2 + y ^ 2) ^ 2 - 2 * 5 * x * (x ^ 2 + y ^ 2) - 5 ^ 2 * y ^ 2 < 0 THEN PSET (320 + x * 20, 240 - y * 20), 1
'
'NEXT y
'NEXT x

'полярные координаты (конопля)
'FOR t = 0 TO 6.28 STEP .001
'r = (1 + SIN(t)) * (1 + .9 * COS(8 * t)) * (1 + .1 * COS(24 * t))
'PSET (320 + r * 50 * COS(t), 240 - 50 * r * SIN(t)), 1
'NEXT t

'полярные кардиоида
'FOR t = 0 TO 6.28 STEP .001
'r = 5 * (1 + COS(t))
'PSET (320 + r * 20 * COS(t), 240 - 20 * r * SIN(t)), 1
'NEXT t

```


Раздел 2. Программное обеспечение ЭВМ. Интернет.

Тема 2.1. Операционные системы (лекция – дискуссия 2 часа)

Функции ОС

. Что такое программное обеспечение?

Под *программным обеспечением* (Software) понимается совокупность программ, выполняемых вычислительной системой.

К программному обеспечению (ПО) относится также вся область деятельности по проектированию и разработке ПО:

- технология проектирования программ (например, нисходящее проектирование, структурное и объектно-ориентированное проектирование и др.);
- методы тестирования программ [ссылка, ссылка];
- методы доказательства правильности программ;
- анализ качества работы программ;
- документирование программ;
- разработка и использование программных средств, облегчающих процесс проектирования программного обеспечения, и многое другое.

Программное обеспечение – *неотъемлемая часть компьютерной системы*. Оно является логическим продолжением технических средств. Сфера применения конкретного компьютера определяется созданным для него ПО.

Сам по себе компьютер не обладает знаниями ни в одной области применения. Все эти знания сосредоточены в выполняемых на компьютерах программах.

Программное обеспечение современных компьютеров включает миллионы программ — от игровых до научных.

Как классифицируется программное обеспечение?

В первом приближении все программы, работающие на компьютере, можно условно разделить на три категории (рис. 6.1):

1. [прикладные программы](#), непосредственно обеспечивающие выполнение необходимых пользователям работ;
2. [системные программы](#), выполняющие различные вспомогательные функции, например:
 - управление ресурсами компьютера;
 - создание копий используемой информации;
 - проверка работоспособности устройств компьютера;
 - выдача справочной информации о компьютере и др.;
3. [инструментальные программные системы](#), облегчающие процесс создания новых программ для компьютера.

При построении классификации ПО нужно учитывать тот факт, что стремительное развитие вычислительной техники и расширение сферы приложения компьютеров резко ускорили процесс эволюции программного обеспечения.

Если раньше можно было по пальцам перечислить основные категории ПО — операционные системы, трансляторы, пакеты прикладных программ, то сейчас ситуация коренным образом изменилась.

Развитие ПО пошло как вглубь (появились новые подходы к построению операционных систем, языков программирования и т.д.), так и вширь (прикладные программы перестали быть прикладными и приобрели самостоятельную ценность).

Соотношение между требующимися программными продуктами и имеющимися на рынке меняется очень быстро. Даже классические программные продукты, такие, как операционные системы, непрерывно развиваются и наделяются интеллектуальными функциями, многие из которых ранее относились только к интеллектуальным возможностям человека.

Кроме того, появились нетрадиционные программы, классифицировать которые по устоявшимся критериям очень трудно, а то и просто невозможно, как, например, программа — *электронный собеседник*.

На сегодняшний день можно сказать, что более или менее определены следующие группы программного обеспечения:

- [операционные системы и оболочки](#);
- [системы программирования](#) (трансляторы, библиотеки подпрограмм, отладчики и т.д.);
- инструментальные системы;
- [интегрированные пакеты программ](#);
- динамические [электронные таблицы](#);
- системы машинной графики;
- системы управления базами данных ([СУБД](#));
- прикладное программное обеспечение.

Структура программного обеспечения показана на рис. 6.2. Разумеется, эту классификацию нельзя считать исчерпывающей, но она более или менее наглядно отражает направления совершенствования и развития программного обеспечения.

Какие программы называют прикладными?

Прикладная программа — это любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

Например, там, где на компьютер возложена задача контроля за финансовой деятельностью какой-либо фирмы, прикладной будет программа подготовки платежных ведомостей.

Прикладные программы могут носить и общий характер, например, обеспечивать составление и печатание документов и т.п.

В противоположность этому, операционная система или инструментальное ПО не вносят прямого вклада в удовлетворение конечных потребностей пользователя.

Прикладные программы могут использоваться либо автономно, то есть решать поставленную задачу без помощи других программ, либо в составе программных комплексов или пакетов.

Какова роль и назначение системных программ?

Системные программы выполняются вместе с прикладными и служат для управления ресурсами компьютера — центральным процессором, памятью, вводом-выводом.

Это программы общего пользования, которые предназначены для всех пользователей компьютера. Системное программное обеспечение разрабатывается так, чтобы компьютер мог эффективно выполнять прикладные программы.

Рис. 6.2. Структура программного обеспечения компьютера

Среди десятков тысяч системных программ особое место занимают операционные системы, которые обеспечивают управление ресурсами компьютера с целью их эффективного использования.

Важными классами системных программ являются также программы вспомогательного назначения — утилиты (лат. *utilitas* — польза). Они либо расширяют и дополняют соответствующие возможности операционной системы, либо решают самостоятельные важные задачи. Кратко опишем некоторые разновидности утилит:

- программы контроля, тестирования и диагностики, которые используются для проверки правильности функционирования устройств компьютера и для обнаружения неисправностей в процессе эксплуатации; указывают причину и место неисправности;
- программы-драйверы, которые расширяют возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д.; с помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся;
- программы-упаковщики (архиваторы), которые позволяют записывать информацию на дисках более плотно, а также объединять копии нескольких файлов в один архивный файл;

- антивирусные программы, предназначенные для предотвращения заражения компьютерными вирусами и ликвидации последствий заражения вирусами;

Компьютерный вирус — это специально написанная небольшая по размерам программа, которая может "приписывать" себя к другим программам для выполнения каких-либо вредных действий — портит файлы, "засоряет" оперативную память и т.д.

- программы оптимизации и контроля качества дискового пространства ;
- программы восстановления информации, форматирования, защиты данных ;
- коммуникационные программы, организующие обмен информацией между компьютерами;
- программы для управления памятью, обеспечивающие более гибкое использование оперативной памяти;
- программы для записи [CD-ROM](#), [CD-R](#) и многие другие.

Часть утилит входит в состав операционной системы, а другая часть функционирует независимо от нее, т.е. автономно.

Что такое операционная система?

Операционная система — это комплекс взаимосвязанных системных программ, назначение которого — организовать взаимодействие пользователя с компьютером и выполнение всех других программ.

Операционная система выполняет роль связующего звена между аппаратурой компьютера, с одной стороны, и выполняемыми программами, а также пользователем, с другой стороны.

Операционная система обычно хранится во внешней памяти компьютера — *на диске*. При включении компьютера она считывается с дисковой памяти и размещается в *ОЗУ*.

Этот процесс называется *загрузкой операционной системы*.

В функции операционной системы входит:

- осуществление диалога с пользователем;
- ввод-вывод и управление данными;
- планирование и организация процесса обработки программ;
- распределение ресурсов (оперативной памяти и кэша, процессора, внешних устройств);
- запуск программ на выполнение;
- всевозможные вспомогательные операции обслуживания;
- передача информации между различными внутренними устройствами;
- программная поддержка работы периферийных устройств (дисплея, клавиатуры, дисковых накопителей, принтера и др.).

Операционную систему можно назвать программным продолжением устройства управления компьютера. Операционная система скрывает от пользователя сложные ненужные подробности взаимодействия с аппаратурой, образуя прослойку между ними. В результате этого люди освобождаются от очень трудоёмкой работы по организации взаимодействия с аппаратурой компьютера.

В зависимости от количества одновременно обрабатываемых задач и числа пользователей, которых могут обслуживать ОС, различают четыре основных класса операционных систем:

1. однопользовательские однозадачные, которые поддерживают одну клавиатуру и могут работать только с одной (в данный момент) задачей;
2. однопользовательские однозадачные с фоновой печатью, которые позволяют помимо основной задачи запускать одну дополнительную задачу, ориентированную, как правило, на вывод информации на печать. Это ускоряет работу при выдаче больших объёмов информации на печать;
3. однопользовательские многозадачные, которые обеспечивают одному пользователю параллельную обработку нескольких задач. Например, к одному компьютеру можно подключить несколько принтеров, каждый из которых будет работать на "свою" задачу;
4. многопользовательские многозадачные, позволяющие на одном компьютере запускать несколько задач нескольким пользователям. Эти ОС очень сложны и требуют значительных машинных ресурсов.

В различных моделях компьютеров используют операционные системы с разной архитектурой и возможностями. Для их работы требуются разные ресурсы. Они предоставляют разную степень сервиса для программирования и работы с готовыми программами.

Операционная система для персонального компьютера, ориентированного на профессиональное применение, должна содержать следующие основные компоненты:

- программы управления вводом/выводом;
- программы, управляющие [файловой системой](#) и планирующие задания для компьютера;
- процессор командного языка, который принимает, анализирует и выполняет команды, адресованные операционной системе.

Каждая операционная система имеет свой командный язык, который позволяет пользователю выполнять те или иные действия:

- обращаться к [каталогу](#);
- выполнять разметку внешних носителей;
- запускать программы;
- другие действия.

Анализ и исполнение команд пользователя, включая загрузку готовых программ из файлов в оперативную память и их запуск, осуществляет командный процессор операционной системы.

Для управления внешними устройствами компьютера используются специальные системные программы — драйверы. Драйверы стандартных устройств образуют в совокупности базовую систему ввода-вывода ([BIOS](#)), которая обычно заносится в постоянное ЗУ компьютера.

Что такое файловая система ОС?

Файл (англ. *file*, папка) — это место постоянного хранения информации: программ, данных для их работы, текстов, закодированных изображений, звуков и др.

Файловая система — это средство для организации хранения файлов на каком-либо носителе.

Файлы физически реализуются как *участки памяти на внешних носителях* — магнитных дисках или CD-ROM.

Каждый файл занимает некоторое количество блоков дисковой памяти. Обычная длина блока — 512 байт.

Обслуживает файлы специальный модуль операционной системы, называемый *драйвером файловой системы*. Каждый файл имеет имя, зарегистрированное в *каталоге* — оглавлении файлов.

Каталог (иногда называется *директорией* или *папкой*) доступен пользователю через командный язык операционной системы.

Его можно просматривать, переименовывать зарегистрированные в нем файлы, переносить их содержимое на новое место и удалять.

Каталог может иметь собственное имя и храниться в другом каталоге наряду с обычными файлами: так образуются иерархические файловые структуры. Пример такой структуры — на рис. 6.3.

Что происходит, когда пользователь подает операционной системе команду "*открыть файл ...*", в которой указано *имя файла* и *имя каталога*, в котором размещён этот файл?

Для выполнения этой команды драйвер файловой системы обращается к своему *справочнику*, выясняет, какие блоки диска соответствуют указанному файлу, а затем передает запрос на считывание этих блоков драйверу диска.

При выполнении команды "*сохранить файл*" драйвер файловой системы ищет на диске незанятые блоки, отмечает их, как распределённые для вновь созданного файла, и передаёт драйверу диска запрос на запись в эти блоки данных пользователя.

Драйвер файловой системы обеспечивает доступ к информации, записанной на магнитный диск, по имени файла и распределяет пространство на магнитном диске между файлами.

Для выполнения этих функций драйвер файловой системы хранит на диске не только информацию пользователя, но и свою собственную служебную информацию. В служебных областях диска хранится список всех файлов и каталогов, а также различные дополнительные справочные таблицы, служащие для повышения скорости работы драйвера файловой системы.

К файловой системе имеет доступ также и любая прикладная программа, для чего во всех языках программирования имеются специальные процедуры.

Понятие файла может быть обращено на любой источник или потребитель информации в машине, например, в качестве файла для программы могут выступать *принтер, дисплей, клавиатура* и др.

Структура файловой системы и структура хранения данных на внешних магнитных носителях определяет удобство работы пользователя, скорость доступа к файлам и т.д.

Какова структура операционной системы MS DOS?

Операционная система MS DOS (Microsoft Disk Operating System) — самая распространенная ОС на 16-разрядных персональных компьютерах. Она состоит из следующих основных модулей (рис. 6.4):

- базовая система ввода/вывода (BIOS);
- блок начальной загрузки (Boot Record);
- модуль расширения базовой системы ввода/вывода (IO.SYS);
- модуль обработки прерываний (MSDOS.SYS);
- командный процессор (COMMAND.COM);
- утилиты MS DOS.

Каждый из указанных модулей выполняет определенную часть функций, возложенных на ОС. Места постоянного размещения этих модулей различны. Так, базовая система ввода/вывода находится в постоянном запоминающем устройстве (ПЗУ), а не на дисках, как все остальные модули.

Базовая система ввода/вывода (BIOS) выполняет наиболее простые и универсальные услуги операционной системы, связанные с осуществлением *ввода-вывода*. В функции BIOS входит также *автоматическое тестирование основных аппаратных компонентов* (оперативной памяти и др.) при включении машины и *вызов блока начальной загрузки DOS*.

Блок начальной загрузки (или просто *загрузчик*) — это очень короткая программа, единственная функция которой заключается в считывании с диска в оперативную память двух других частей DOS — модуля расширения базовой системы ввода/вывода и модуля обработки прерываний.

Модуль расширения базовой системы ввода/вывода дает возможность использования *дополнительных драйверов*, обслуживающих новые внешние устройства, а также *драйверов для нестандартного обслуживания внешних устройств*.

Модуль обработки прерываний реализует основные высокоуровневые услуги DOS, поэтому его и называют основным.

Командный процессор DOS обрабатывает команды, вводимые пользователем.

Утилиты DOS — это программы, поставляемые вместе с операционной системой в виде отдельных файлов. Они выполняют действия обслуживающего характера, например, разметку дисков, проверку дисков и т.д.

Что такое программы-оболочки?

Оболочки — это программы, созданные для упрощения работы со сложными программными системами, такими, например, как DOS. Они преобразуют неудобный командный пользовательский интерфейс в дружелюбный графический интерфейс или интерфейс типа "меню". Оболочки предоставляют пользователю удобный доступ к файлам и обширные сервисные услуги.

Самая популярная у пользователей IBM-совместимого ПК оболочка — пакет программ *Norton Commander*. Он обеспечивает:

- создание, копирование, пересылку, переименование, удаление, поиск файлов, а также изменение их атрибутов;
- отображение дерева каталогов и характеристик входящих в них файлов в форме, удобной для восприятия человека;
- создание, обновление и распаковку архивов (групп сжатых файлов);
- просмотр текстовых файлов;
- редактирование текстовых файлов;
- выполнение из её среды практически всех команд DOS;
- запуск программ;
- выдачу информации о ресурсах компьютера;
- создание и удаление каталогов;
- поддержку межкомпьютерной связи;
- поддержку электронной почты через модем.

ОС семейства windows

В начале 90-х годов во всем мире огромную популярность приобрела графическая оболочка *MS-Windows 3.x*, преимущество которой состоит в том, что она облегчает использование компьютера, и её графический интерфейс вместо набора сложных команд с клавиатуры позволяет выбирать их мышью из меню практически мгновенно. Операционная среда Windows, работающая совместно с операционной системой DOS, реализует все свойства, необходимые для производительной работы пользователя, в том числе – многозадачный режим.

Оболочка Norton Navigator — это набор мощных программ для управления файлами, расширяющий возможности Windows. Позволяет экономить время практически на всех операциях: поиск файлов, копирование и перемещение файлов, открытие каталогов. Что собой представляют операционные системы Windows NT и Windows 95?

Windows NT (*NT* — англ. *New Technology*) — это операционная система, а не просто графическая оболочка. Она использует все возможности новейших моделей персональных компьютеров и работает без DOS.

Windows NT — 32-разрядная ОС со встроенной сетевой поддержкой и развитыми многопользовательскими средствами. Она предоставляет пользователям истинную многозадачность, многопроцессорную поддержку, секретность, защиту данных и многое другое.

Эта операционная система очень удобна для пользователей, работающих в рамках локальной сети, для коллективных пользователей, особенно для групп, работающих над большими проектами и обменивающимися данными.

Windows 95 представляет собой универсальную высокопроизводительную многозадачную и многопоточную 32-разрядную ОС нового поколения с графическим интерфейсом и расширенными сетевыми возможностями.

Windows 95 — интегрированная среда, обеспечивающая эффективный обмен информацией между отдельными программами и предоставляющая пользователю широкие возможности работы с мультимедиа, обработки текстовой, графической, звуковой и видеoinформации.

Интегрированность подразумевает также *совместное использование ресурсов компьютера всеми программами*.

Эта операционная система *обеспечивает работу пользователя в сети*, предоставляя встроенные средства поддержки для обмена файлами и меры по их защите, возможность совместного использования принтеров, факсов и других общих ресурсов. Windows 95 позволяет отправлять сообщения электронной почтой, факсимильной связью, поддерживает удаленный доступ.

Применяемый в *Windows 95* защищенный режим не позволяет прикладной программе в случае сбоя нарушить работоспособность системы, надежно предохраняет приложения от случайного вмешательства одного процесса в другой, обеспечивает определенную устойчивость к вирусам.

Пользовательский интерфейс *Windows 95* прост и удобен.

В отличие от оболочки *Windows 3* эта операционная система не нуждается в установке на компьютере операционной системы DOS. Она предназначена для установки на настольных ПК и компьютерах блокнотного типа с процессором 486 или Pentium.

Рекомендуемый размер оперативной памяти 8-16 Мбайт.

После включения компьютера и выполнения тестовых программ BIOS операционная система *Windows 95* автоматически загружается с жесткого диска. После загрузки и инициализации системы на экране появляется *рабочий стол*, на котором размещены различные гра-

фические объекты. Пользовательский интерфейс спроектирован так, чтобы максимально облегчить усвоение этой операционной системы новичками и создать комфортные условия для пользователя.

Что такое транслятор, компилятор, интерпретатор?

Транслятор (англ. *translator* — переводчик) — это программа-переводчик. Она преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Трансляторы реализуются в виде компиляторов или интерпретаторов. С точки зрения выполнения работы компилятор и интерпретатор существенно различаются.

Компилятор (англ. *compiler* — составитель, собиратель) читает всю программу *целиком*, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Интерпретатор (англ. *interpreter* — истолкователь, устный переводчик) переводит и выполняет программу *строка за строкой*.

После того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны. В то же время программа, обрабатываемая интерпретатором, должна заново *переводиться* на машинный язык при каждом очередном запуске программы.

Откомпилированные программы работают быстрее, но *интерпретируемые* проще исправлять и изменять.

Каждый конкретный язык ориентирован либо на компиляцию, либо на интерпретацию — в зависимости от того, для каких целей он создавался. Например, *Паскаль* обычно используется для решения довольно сложных задач, в которых важна скорость работы программ. Поэтому данный язык обычно реализуется с помощью *компилятора*.

С другой стороны, *Бейсик* создавался как язык для начинающих программистов, для которых построчное выполнение программы имеет неоспоримые преимущества.

Иногда для одного языка имеется *и компилятор, и интерпретатор*. В этом случае для разработки и тестирования программы можно воспользоваться интерпретатором, а затем откомпилировать отлаженную программу, чтобы повысить скорость ее выполнения.

Что такое системы программирования?

Система программирования — это система для разработки новых программ на конкретном языке программирования.

Современные системы программирования обычно предоставляют пользователям мощные и удобные средства разработки программ. В них входят:

- [компилятор](#) или [интерпретатор](#);
- интегрированная среда разработки;
- средства создания и редактирования текстов программ;
- обширные [библиотеки стандартных программ](#) и функций;
- [отладочные программы](#), т.е. программы, помогающие находить и устранять ошибки в программе;
- "дружественная" к пользователю диалоговая среда;
- многооконный режим работы;
- мощные графические библиотеки; [утилиты](#) для работы с библиотеками
- встроенный [ассемблер](#);
- встроенная справочная служба;
- другие специфические особенности.

Популярные системы программирования – *Turbo Basic, Quick Basic, Turbo Pascal, Turbo C*.

В последнее время получили распространение системы программирования, ориентированные на создание *Windows-приложений*: Borland Delphi 3.0

- пакет *Borland Delphi* (Дельфи) — блестящий наследник семейства компиляторов Borland Pascal, предоставляющий качественные и очень удобные средства визуальной разработки. Его исключительно быстрый компилятор позволяет эффективно и быстро решать практически любые задачи прикладного программирования.
- пакет *Microsoft Visual Basic* — удобный и популярный инструмент для создания Windows-программ с использованием визуальных средств. Содержит инструментарий для создания *диаграмм* и *презентаций*.
- пакет *Borland C++* — одно из самых распространённых средств для разработки DOS и Windows приложений.

Тема 2.2. Текстовый редактор (лекция – дискуссия 6 часов)

Функции текстового редактора

Текстовый редактор — самостоятельная [компьютерная программа](#) или компонент программного комплекса (например, [редактор исходного кода интегрированной среды разработки](#) или окно ввода в браузере), предназначенная для создания и изменения [текстовых данных](#) в общем и [текстовых файлов](#) в частности^[1].

Текстовые редакторы предназначены для работы с текстовыми файлами в интерактивном режиме. Они позволяют просматривать содержимое текстовых файлов и производить над ними различные действия — вставку, удаление и копирование текста, контекстный поиск и замену, сортировку строк, просмотр кодов символов и конвертацию [кодировок](#), печать и т. п.

Часто интерактивные текстовые редакторы содержат дополнительную функциональность, призванную автоматизировать действия по редактированию (от записываемых последовательностей нажатий клавиш до полноценных встроенных [языков программирования](#)), или отображают текстовые данные специальным образом (например, с [подсветкой синтаксиса](#)).

Многие текстовые редакторы являются [редакторами исходного кода](#), то есть они ориентированы на работу с [текстами программ](#) на тех или иных [компьютерных языках](#).

Виды текстовых редакторов

Построчный (строковый) текстовый редактор (англ. *line editor*) работает с текстом как последовательностью пронумерованных строк, выполняя операции над текстом в указанных строках^[2]. Примером такого редактора может быть edlin, входивший в состав [MS-DOS](#).

Контекстный (строковый) редактор (англ. *context editor*), примером которого может быть ECCE (англ. *Edinburgh Compatible Context Editor*)^[3], выполняет операции над текстом в текущей позиции.

Экранный текстовый редактор позволяет пользователю перемещать [курсор](#) в тексте с помощью клавиш или других устройств ввода^[4].

Текстовые процессоры

Основная статья: [Текстовый процессор](#)

Текстовые процессоры ориентированы на оформление и форматирование текстов и внедрение в них сторонних объектов (шрифтов, таблиц, формул, графиков т. п.) и характеризуются наличием WYSIWYG-режимов. Поскольку в текстовом формате не предусмотрено хранение информации об оформлении текста, текстовые процессоры работают либо с файлами, в которых тексты представлены в оформлении какого-либо [языка разметки](#) вроде [HTML](#), либо с файлами в собственных «[двоичных](#)» форматах.

Текстовый редактор Microsoft Word

РАБОТА В WORD XP

Обработка текста на компьютере осуществляется с помощью специальных программ, называемых *текстовыми процессорами*. С их помощью можно ввести и отформатировать текст, исправить ошибки и просмотреть документ перед печатью.

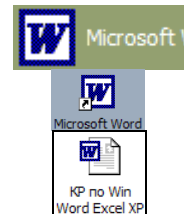
Microsoft Word XP представляет собой текстовый процессор, который используется для подготовки личной и деловой документации. Помимо стандартных средств Word XP включает многие функции настольных издательских систем, а также шаблоны типовых документов, на основе которых можно легко и быстро создавать документы, имеющие профессиональный вид.

ЗАГРУЗКА WORD XP

☞ **Выполнение задания.** Запустите приложение Word XP.

- Нажмите кнопку [ПУСК](#) на Панели задач / Все программы и выберите левой кнопкой мыши команду Microsoft Word или

- Нажмите кнопку **ПУСК** на Панели задач / щелкните по значку Microsoft Word в левом столбце Панели
- Дважды щелкните по ярлычку запуска Word XP или
- Дважды щелкните по значку ранее созданного пустого документа Microsoft Word XP, или
- Чтобы открыть документ, с которым вы недавно работали, нажмите кнопку **Пуск**,



остановите указатель на пункте Недавние документы и щелкните мышью по документу, который желательно запустить.

НАСТРОЙКА ВИДА ДОКУМЕНТА WORD XP

Выполнение задания.

- Щелкните по кнопке **Развернуть** в зоне заголовка окна, если после вызова программы ее окно не занимает всего экрана. Также сделайте полноэкранным окно документа.
- Убедитесь, что ваш документ имеет вид разметки страниц (рис. 2.1). Вид / Разметка страниц.
- А так же в наличии линейки на экране (рис. 2.1). Если линейка отсутствует, выполните команду Вид / Линейка (слева должна стоять галочка).
- С помощью команды Вид / Панели инструментов выведите панели инструментов СТАНДАРТНАЯ и ФОРМАТИРОВАНИЕ (слева от них должна стоять галочка), а лишние уберите.

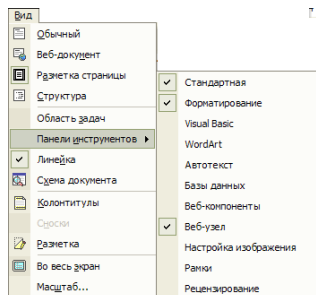


Рис.2.1. Меню пункта Вид

- Установите параметры страницы. Файл / Параметры страницы (рис. 2.2.): Верхнее – 2,5 см., Нижнее – 2,5 см., Левое – 2,5 см., Правое – 1,5 см. (Поля меняются с помощью стрелок выбора



-) и нажмите кнопку **ОК**.
- Для установки параметров страницы также можно дважды щелкнуть в области вертикальной линейки (при режиме экрана Разметка).
- Перейдите на вкладку Размер бумаги (рис.2.2). Проверьте, чтобы лист имел размеры Ширина - 21 см, Высота - 29,7 см. Если это не так, то можете выбрать в окне Размер

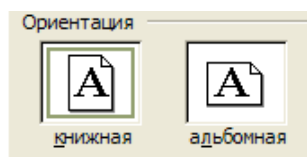
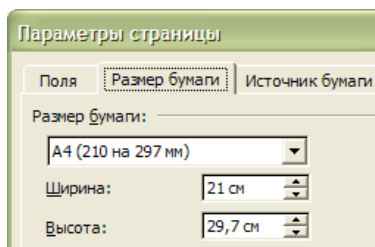
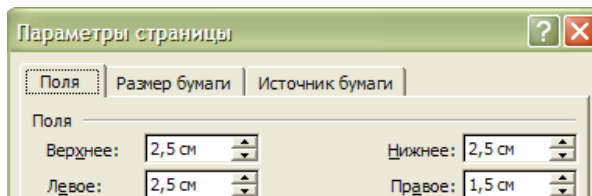


Рис.2.2. Фрагменты диалогового окна Параметры страницы

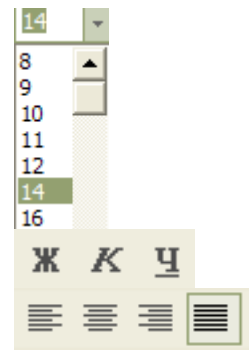
- бумаги формат А4 210 * 297 мм., или установите параметры с помощью стрелок выбора в окнах Ширина и Высота.
- Проверьте, чтобы ориентация листа была установлена как книжная.
- Нажмите кнопку **ОК**.
- Включите кнопку **Непечатаемые символы**. Они помогут Вам в дальнейшем редактировать текст.



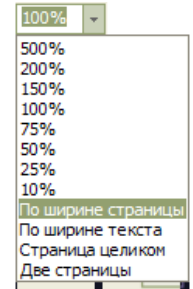
- Установите шрифт Times New Roman, размером 14 пунктов, с выравниванием по ширине с помощью команды **Формат / Шрифт** (рис. 2.7) или с помощью панели инструментов **Форматирование** (рис.2.8).
- Раскройте окно со списком шрифтов и выберите шрифт с именем Times New Roman.



- В соседнем окне Размер шрифтов задайте размер, равный 14 пунктов.

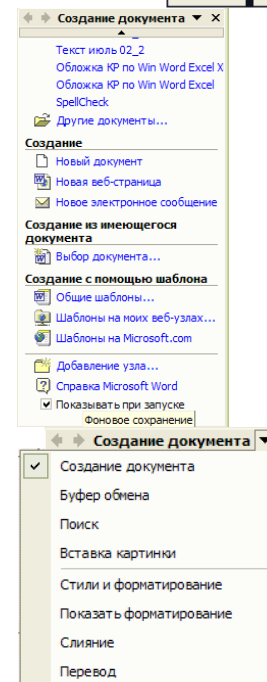


- Пока не трогайте кнопки Полужирный, Курсив, Подчеркнутый.
- Выравнивание строки установите нажатием на кнопку По ширине.
- Несколько раз нажмите на клавишу клавиатуры **Enter**. Этим самым, Вы установите для себя несколько строк с заданным форматированием.
- Установите масштаб экрана Вид / Масштаб / По ширине страницы и нажмите **ОК**.
- Если размер документа нужно изменить, то попробуйте щелкнуть в окне Масштаб и экспериментально определите необходимый масштаб экрана, завершая ввод чисел нажатием на клавишу **Enter**.



☛ **Примечание.** Панель задачи, которая открывается рядом с окном документа, содержит команды, характерные для ряда часто повторяющихся задач. Так, на рис. 2.3. изображена панель задачи Создание документа, появляющаяся при задании команды Создать. Она позволяет открыть несколько последних документов Word XP, создать новый документ на основе существующего документа или шаблона.

Рис.2.3. Фрагмент панели задачи команды Создание документа



Чтобы отобразить панель другой задачи, щелкните на стрелке Другие области задач и выберите нужную (рис. 2.4). Если панель задачи не нужна, можно убрать ее с экрана, освободив тем самым место для работы. Для этого в меню Вид щелкните на команде Область задач. Чтобы вернуть панель задачи на экран, снова щелкните на этой команде.

Рис.2.4. Панель Другие области задачи

СОХРАНЕНИЕ СОЗДАННОГО ДОКУМЕНТА

Текст, который отображается в окне документа, хранится в оперативной памяти компьютера. При завершении работы с Word XP он будет утерян. Поэтому, чтобы сохранить введенный текст, нужно записать документ в файл на жесткий диск компьютера. Тогда его можно будет открыть позже и продолжить работу.

☛ **Выполнение задания.**

- Выполните команду Файл / Сохранить как... / выберите свою папку, в которую помещается документ (Например, Рабочий стол\ИМЯ ГРУППЫ) / присвойте документу имя Контрольная работа. Нажмите кнопку **Сохранить** (рис 2.5.).
- Далее через некоторые промежутки времени сохраняйте документ по команде Файл / Сохранить или нажимайте кнопку **Сохранить** на панели инструментов Стандартная.

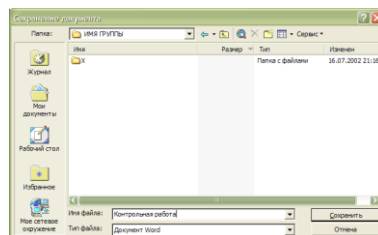


Рис.2.5. Диалоговое окно Сохранение документа

☛ **Примечание.** В этом случае новая версия документа замещает предыдущую в том же самом файле.

В процессе работы Word XP автоматически сохраняет документ, чтобы восстановить его в случае зависания программы или отключения питания компьютера.

Чтобы включить функцию автоматического сохранения и указать нужный интервал:

- Выделите команду Сервис / Параметры / на вкладке Сохранение выделите флажок Автосохранение каждые / введите значение интервала в минутах / щелкните на кнопке **ОК**.
- При аварийном отключении компьютера (что трижды произошло при написании этого фрагмента текста и больше ни разу не повторялось) Word XP позволяет выполнить восстановление документов (рис. 2.6).

- Откройте Word XP. Просмотрите список файлов в области задач Восстановление документов и решите, какие из них следует оставить.
- В области задач Восстановление документов наведите указатель на имя каждого файла, который требуется сохранить, щелкните стрелку рядом с именем файла, а затем выполните одно из следующих действий:
- Чтобы начать работу с файлом, выберите команду Открыть;
- Чтобы сохранить файл, выберите команду Сохранить как (рис. 2.5), а затем введите имя для файла. По умолчанию файл сохраняется в той же папке, что и исходный файл. При использовании имени исходного файла исходный файл заменяется.
- Открыв или сохранив все файлы, которые требуется оставить, нажмите кнопку Закрывать в области задач Восстановление документов.

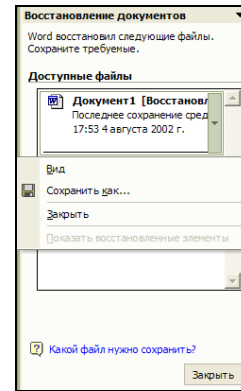
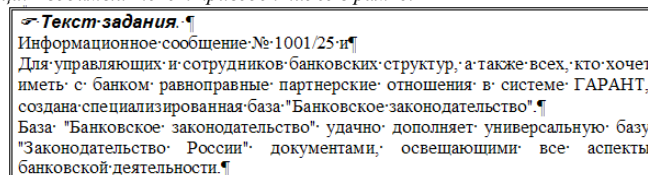


Рис.2.6. Диалоговое окно области задач Восстановление документов

РАБОТА С ОСНОВНЫМ ТЕКСТОМ

Выполнение задания.

Наберите с клавиатуры основной текст без разбивки на колонки. Между словами должен быть только один знак пробела. Клавиша **Enter** нажимается только в конце абзаца. Вводимый текст приведен ниже в рамке.



ПЕРЕМЕЩЕНИЕ ПО ТЕКСТУ И ВЫДЕЛЕНИЕ ТЕКСТА

Для перемещения по тексту можно использовать клавиши-стрелки, но более целесообразно это делать при одновременном нажатии клавиши **Ctrl**:

- **Правая стрелка** - на одно слово вперед
- **Левая стрелка** - на одно слово назад
- **Стрелка вверх** - к началу абзаца
- **Стрелка вниз** - к началу следующего абзаца
- **Home** - в начало документа
- **End** - в конец документа

Просто нажимая соответствующие клавиши, вы можете добиться следующего эффекта:

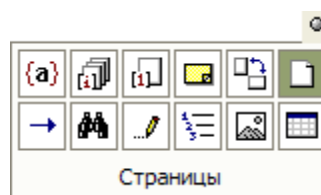
- **Home** – в начало строки, **End** – в конец строки.
- **PgUp** - На экран вверх (точка ввода не изменяется)
- **PgDn** - На экран вниз (точка ввода не изменяется)

Для редактирования текста можно использовать клавиши:

- **Backspace** удаляет символ, расположенный слева от курсора.
- **Delete (Del)** удаляет символ, следующий за курсором.

Для перемещения по документу также можно воспользоваться [вертикальной и горизонтальной полосами прокрутки](#) (п. 1.6.1.2.4).. При этом изменяется лишь отображение документа в окне, а курсор остается на месте. Положение курсора отображается в [Строке состояния](#) (п. 1.6.1.1.4).

Для просмотра длинных документов можно воспользоваться кнопкой Выбор объекта перехода внизу вертикальной полосы прокрутки. Если щелкнуть по этой кнопке, откроется меню, содержащее способы просмотра документа по страницам, примечаниям, рисункам и другим объектам.



Чтобы выделить фрагменты текста:

- **Одного слова** - дважды щелкните внутри слова.
- **Нескольких слов** - протащите указатель мыши с нажатой левой кнопкой по отмечаемым словам.
- **Одной строки** - поместите курсор слева от выделяемой строки, чтобы он принял вид стрелки, наклоненной вправо, и щелкните мышью. Для выбора нескольких строк щелкните слева от первой из них и протяните курсор вверх или вниз.
- **Предложений** - удерживая клавишу **Ctrl**, щелкните мышью внутри предложения
- **Абзаца** - выполните два щелчка слева от него или три щелчка внутри него. Для выделения нескольких абзацев выделите первый и, не отпуская кнопку мыши, переместите указатель вверх или вниз.
- **Всего документа целиком** - произведите тройной щелчок мышью слева от текста или нажмите **Ctrl + A**.
- Для выделения **прямоугольного фрагмента текста** нажмите клавишу **Alt** и, удерживая ее, перемещайте курсор мыши по тексту.

Для отмены выделения текста, щелкните мышью где-нибудь внутри него или нажмите любую клавишу - стрелку.

Для удаления выбранного текста нажмите клавишу **Del** или **Backspace**.

Для восстановления удаленного текста сразу после завершения операции нажмите кнопку Отменить Стандартной панели инструментов.

Для возвращения отмененных команд нажмите кнопку Вернуть на Стандартной панели инструментов.

ФОРМАТИРОВАНИЕ ТЕКСТА.

Команды форматирования текста объединены в пункте меню ФОРМАТ (рис. 2.7.)

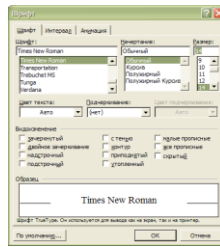


Рис.2.7. Диалоговое окно Шрифт команды меню Формат

Часть из них выведена на панель инструментов Форматирование (рис. 2.8)

☞ **Выполнение задания.**

Выделите заголовок (п. 2.4)

- Для выравнивания его по центру нажмите на Панели инструментов Форматирование кнопку **По центру**.

- Для того чтобы оформить заголовок жирным шрифтом и сделать все буквы прописными, выполните следующие действия: не снимая выделения с заголовка, выполните команду Формат / Шрифт... / выберите полужирное начертание и установите флажок в поле Все прописные / ОК.

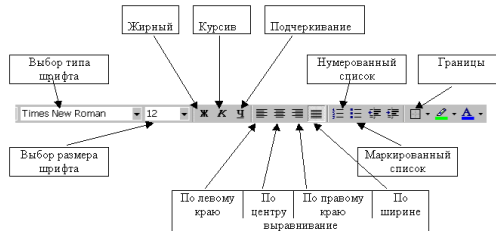
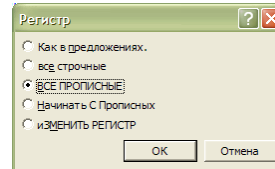


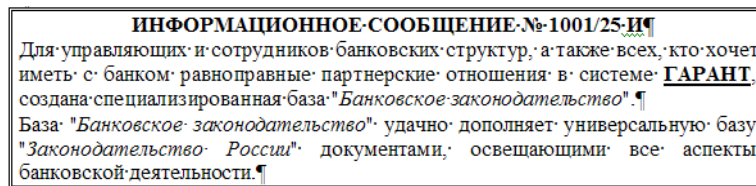
Рис.2.8. Кнопки панели инструментов Форматирование

- Для того чтобы текст заголовка был напечатан главными буквами, выделите команду Формат / Регистр и установите переключатель в положение ВСЕ ПРОПИСНЫЕ / нажмите кнопку ОК.



- Основной текст, после выделения фрагментов, (п.2.4.1) форматируется аналогично с помощью пункта меню Формат / Шрифт ... или кнопок на панели инструментов:

- Основной текст выравнивается по ширине с помощью кнопки По ширине на панели инструментов
- В результате форматирования текст примет следующий вид:



Текст служебной записки чаще всего форматируется в виде сплошных строк, расположенных на странице от края до края. Для коротких документов такое расположение вполне приемлемо. Однако длинные документы с мелким шрифтом будут читаться гораздо легче, если использовать две или три узкие колонки (столбцы). Чем меньше шрифт, тем больше колонок можно использовать.

- Для того чтобы разделить текст на колонки, необходимо выделить преобразуемый текст. Начинайте, выделять с нижней строки тек-



ста, не захватывая последний символ абзаца

- После того, как выделили текст, выполняем команду Формат / Колонки... (рис. 2.9). В диалоговом окне Колонки выбираем число колонок: две / если это необходимо в задании, то устанавливаете флажок Разделитель / ОК.

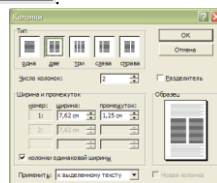


Рис.2.9. Диалоговое окно Колонки

- Для создания буквицы выделяем первую букву абзаца и выполняем команду Формат / Буквица.
- В диалоговом окне Буквица (рис. 2.10) выбираем положение в тексте, высота в строках – 2 / ОК.

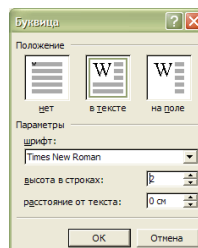


Рис.2.10. Диалоговое окно Буквица

дополняет универсальную
России" документами, осв
банковской деятельности.

☛ **Примечание.** Буквица и колонки конфликтуют между собой, поэтому вначале создайте колонки и только потом – буквицу. Текст примет следующий вид:

ИНФОРМАЦИОННОЕ СООБЩЕНИЕ № 1001/25-ИЧ

<p>Для управляющих и сотрудников банковских структур, а также всех, кто хочет иметь с банком равноправные партнерские отношения в системе ГАРАНТ, создана специализированная база "Банковское законодательство".</p>	<p>База "Банковское законодательство" удачно дополняет универсальную базу "Законодательство России" документами, освещающими все аспекты банковской деятельности.</p>
---	---

Списки применяются в тех случаях, когда нужно организовать или упорядочить информацию, представляющую собой перечень элементов. Списки бывают двух видов: **маркированные** и **нумерованные**. В маркированных списках каждый элемент помечается декоративным символом – маркером. Нумерованные списки используются в том случае, когда нужно подчеркнуть число или последовательность элементов. При удалении, перемещении или вставке элементов нумерованного списка Word XP обновляет нумерацию. Если элементы списка расположены не в том порядке, который требуется, можно отсортировать выделенный список по возрастанию или убыванию командой Таблица / Сортировка.

Word XP включает несколько стандартных форматов списков. При желании можно создать собственный стиль или воспользоваться рисунками и символами.

Так как первый абзац начинается с буквицы, то создадим список только из одного второго абзаца.

- Щелкните мышью в любом месте второго абзаца (или выделите его полностью).
- На панели инструментов Форматирование щелкните на кнопке Нумерация. Выделенный текст будет преобразован в нумерованный



список.

- На панели инструментов Форматирование щелкните на кнопке Маркеры. Выделенный текст будет преобразован в маркированный



список

- На панели инструментов Форматирование щелкните на кнопкам Уменьшить отступ или Увеличить отступ.



Маркированный список сместится влево или вправо.

☛ **Примечание.** Для изменения внешнего вида списка можно воспользоваться командой Формат / Список... Затем выбрать необходимые установки. В свободное время можете поэкспериментировать.

Окончательный вид текста будет следующий:

ИНФОРМАЦИОННОЕ СООБЩЕНИЕ № 1001/25-ИЧ

<p>Для управляющих и сотрудников банковских структур, а также всех, кто хочет иметь с банком равноправные партнерские отношения в системе ГАРАНТ, создана специализированная база "Банковское законодательство".</p>	<p>•→ База "Банковское законодательство" удачно дополняет универсальную базу "Законодательство России" документами, освещающими все аспекты банковской деятельности.</p>
---	--

СОЗДАНИЕ КОЛОНТИТУЛА (вставка в него рисунка, фигурного текста и надписи)

СОЗДАНИЕ КОЛОНТИТУЛА

Колонтитулом называется фрагмент текста (может включать в себя рисунки и другие объекты), который, будучи введенным один раз, появляется на каждой странице многостраничного документа. Колонтитул может быть верхним и нижним (могут быть оба одновременно). Создайте колонтитул следующего вида



☛ **Выполнение задания.**

- Для входа в колонтитул выделите команду Вид / Колонтитулы. Нажмите несколько раз клавишу Enter для того, чтобы размер колонтитула по высоте увеличился приблизительно до трех сантиметров. Вводимый далее фигурный текст, рисунок и таблица с реквизитами, не должны превышать по высоте эти размеры. После того как Вы попадаете в Колонтитул, появляется дополнительная панель инструментов (рис. 2.11.)

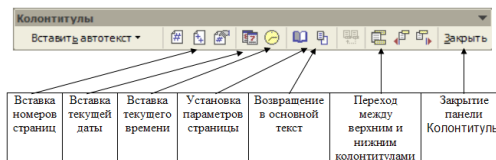


Рис.2.11. Панель инструментов Колонтитулы

РАЗМЕЩЕНИЕ РИСУНКОВ

☛ **Выполнение задания.**

- Чтобы поместить рисунок, выполните команду: Вставка / Рисунок / Картинки / Коллекция картинок... (рис. 2.12)
- Выберите из предложенных рисунков любой, например, такой, что Вы видите справа, и нажмите кнопку **ВСТАВИТЬ**.



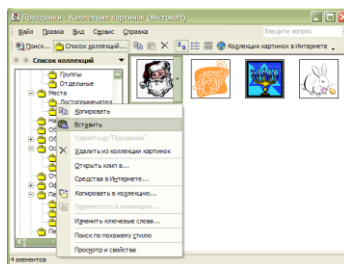


Рис.2.12. Диалоговое окно Коллекция картинок (подраздел Праздники)

☛ **Примечание.** Иногда эта операция не проходит, (что и случилось в момент написания данного пособия), тогда можно щелкнуть правой кнопкой мыши на рисунке, выбрать команду Копировать, а затем вставить как объект в нужном месте. Результат будет тот же.

- Для того чтобы изменить размер рисунка, щелкните по нему. Вокруг рисунка появятся квадратики - маркеры изменения размера. Размер объекта изменяется путем перетаскивания этих маркеров.
- На следующем этапе вам надо будет переместить созданный рисунок к правому краю колоннитула. Для этого можно просто установить выравнивание по правому краю, предварительно выделив его.

Но если вам понадобится произвольное перемещение рисунка (фигурного текста, надписи и так далее), то выполните следующие операции:

- Щелкните правой кнопкой мыши по рисунку. В появившемся контекстном меню выберите команду **Формат рисунка...**
- В появившемся диалоговом окне **Формат рисунка** (рис. 2.13) откройте вкладку **Положение** и выберите **Обтекание вокруг рамки** и установите переключатель **Горизонтальное выравнивание** в положение **другое**.
- Нажмите кнопку **ОК**. Переместить рисунок можно, зацепив его мышкой в любом месте и, не отпуская левой клавишей мыши, переместить его в нужное место рисунка.

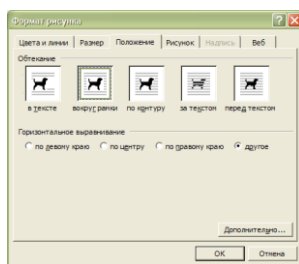


Рис.2.13. Диалоговое окно Формат СОЗДАНИЕ ФИГУРНОГО ТЕКСТА

☛ **Выполнение задания.**

- Для того чтобы создать логотип предприятия, необходимо вставить фигурный текст: **Вставка / Рисунок / Объект WordArt...** (рис. 2.14).

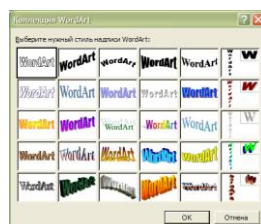


Рис.2.14. Диалоговое окно Коллекция WordArt

- Выберите нужный стиль надписи WordArt, нажмите **ОК** и введите необходимый текст / **ОК**.

Чтобы изменить размер и положение объекта, необходимо выполнить аналогичные действия, что и при работе с [рисунком](#) (п. 2.5.2).

ВСТАВКА И РЕДАКТИРОВАНИЕ НАДПИСИ

☛ **Выполнение задания.**

Чтобы ввести между рисунком и фигурным текстом **ИМЯ** группы и собственную фамилию, воспользуйтесь **надписью**:

- Выделите команду **Вставка / Надпись**.
- Курсор превратится в крестик. Нарисуйте им прямоугольник, в котором расположите саму надпись. В дальнейшем размеры рамки надписи и ее местонахождение можно будет **изменить** (п. 2.5.2).
- **Задайте** размер шрифта **20 пт.**, полужирный, выравнивание **по центру** (рис.2.8).
- Напишите **ИМЯ ГРУППЫ**, нажмите клавишу **Enter**, затем напишите **СВОЮ ФАМИЛИЮ**. Щелкните за пределами надписи. Результат может иметь следующий вид (рис. 2.15).

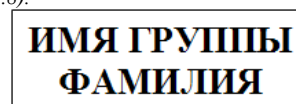


Рис.2.15. Результат ввода надписи до ее форматирования

Займитесь форматированием надписи.

☛ **Примечание.** Учтите, что изученное Вами ниже форматирование надписей, подходит ко многим другим объектам Word XP.

Чтобы изменить цвет символов текста надписи на синий:

- Выделите фрагмент текста, цвет символов которого нужно изменить.
- Щелкните левой кнопкой мыши по стрелке справа от кнопки **Цвет шрифта** панели инструментов **Форматирование**. Появится палитра цветов символов (рис. 2.16).
- Выделите нужный цвет, например, синий.

Измените цвет рамки на индиго, тип ограничивающей линии на двойную непрерывную размером 3 пт:

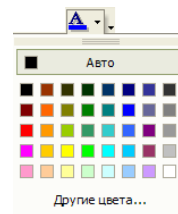
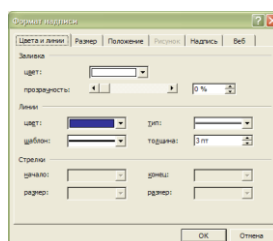
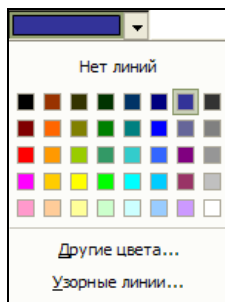


Рис.2.16. Панель Цвет шрифта

- Выделите надпись
- Щелкните правой кнопкой мыши по границе надписи и в появившемся контекстном меню выберите команду **Формат надписи**.
- В диалоговом окне **Формат надписи** (рис. 2.17) раскройте вкладку **Цвета и линии** (рис.2.18)

Рис.2.17. Диалоговое окно Формат надписи



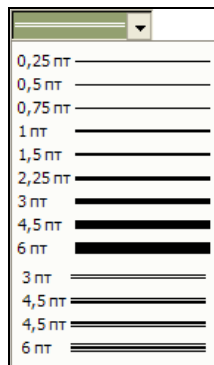


- В палитре Цвет группы Линии выделите цвет линий рамки. Если нажмете кнопку Нет линий в палитре, то будут удалены линии рамки.

Рис.2.18. Вкладка Цвета и линии

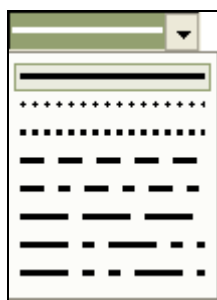
- В палитре Тип группы Линии (рис. 2.19) выделите толщину линий рамки или установите толщину линий, указав значение в поле Толщина.

Рис.2.19. Палитра Тип группы Линии



- В палитре Шаблон группы Линии (рис. 2.20) выделите необходимый стиль представления линии.
- Нажмите кнопку ОК.

Рис.2.20. Палитра Шаблон группы Линии



Чтобы изменить цвет заливки надписи на светло-бирюзовый:

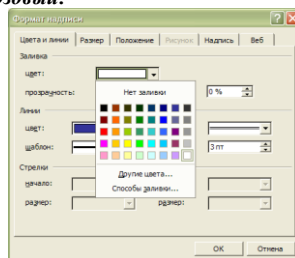


Рис.2.21. Диалоговое окно Формат надписи

- Выделите надпись.
- Щелкните правой кнопкой мыши на границе надписи и в появившемся контекстном меню выберите команду меню Формат надписи.
- В диалоговом окне Формат надписи (рис. 2.21) раскройте вкладку Цвета и линии.
- В палитре Цвет группы Заливки выделите цвет заливки или нажмите кнопку Нет заливки, чтобы удалить заливку.

- Для того чтобы сделать заливку светлее, установите флажок Полупрозрачный.
 - Нажмите кнопку ОК.
- Дополнительные цвета и специальные стили заливки позволяют разнообразить вид надписей.

☞ **Выполнение задания.**

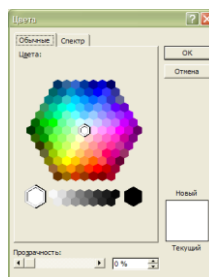
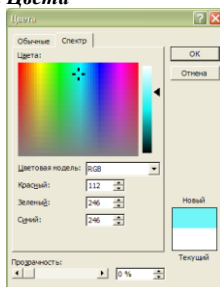
Чтобы выбрать дополнительный цвет для линий или заливки:

- Выделите надпись.
- Вызовите диалоговое окно Цвета Заливки или Линии (рис. 2.18. или 2.21). Для этого выберите команду Другие цвета линий или Другие цвета заливки.

Появится диалоговое окно Цвета (рис. 2.22).

- Чтобы выбрать дополнительный цвет из стандартных цветов Windows, раскройте вкладку Обычные и выделите нужный цвет щелчком мыши. Чтобы снизить интенсивность выбранного цвета, установите флажок Полупрозрачный. Нажмите кнопку ОК

Рис.2.22. Вкладка Обычные диалогового окна Цвета



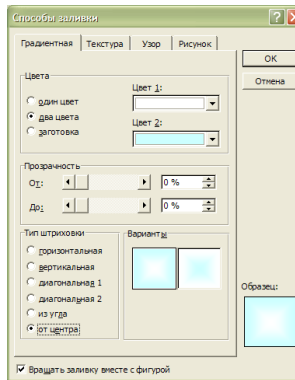
- Если во вкладке Обычные нет подходящего цвета, то раскройте вкладку Спектр, (рис. 2.23) и, создайте свой дополнительный цвет:
- На схеме Цвета выделите нужный цвет при помощи мыши. Используйте регулятор насыщенности справа от схемы Цвета, чтобы настроить насыщенность создаваемого цвета.
- Чтобы снизить интенсивность выбранного цвета, установите флажок Полупрозрачный.
- Нажмите кнопку ОК.

Рис.2.23. Вкладка Спектр диалогового окна Цвета

Чтобы создать градиентную заливку для надписи:

- Раскройте вкладку Градиентная диалогового окна Заливка (рис.2.24).
- В группе Цвета выберите цвет заливки или воспользуйтесь одной из заготовок.
- В группе Тип штриховки выберите тип штриховки.
- Выделите один из предложенных вариантов заливки на схеме Варианты
- Нажмите кнопку ОК.

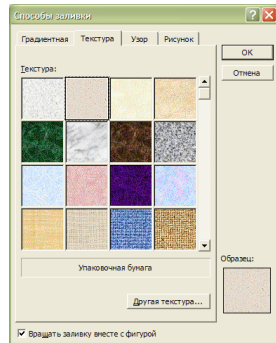
Рис.2.24. Вкладка Градиентная диалогового окна Заливка



Чтобы создать текстурную заливку для надписи:

- Раскройте вкладку Текстура диалогового окна Заливка (рис.2.25).
- Выделите один из 24 встроенных вариантов текстуры или нажмите кнопку Другая текстура и укажите собственный файл с текстурным рисунком.
- Нажмите кнопку ОК.

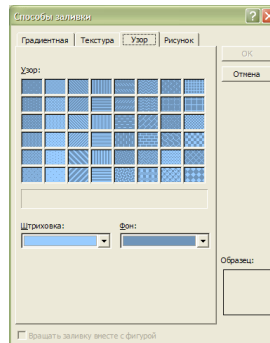
Рис.2.25. Вкладка Текстура диалогового окна Заливка



Чтобы создать заливку надписи с использованием узора:

- Раскройте вкладку Узор.
- Выделите узор заливки на схеме Узор (рис.2.26).
- В раскрывающемся списке Штриховка выделите цвет штриховки узора.
- В раскрывающемся списке Фон выделите цвет фона узора.
- Нажмите кнопку ОК.

Рис.2.26. Вкладка Узор диалогового окна Заливка



Один из вариантов форматирования надписи представлен на рис. 2.27.

Рис.2.27. Окончательные результаты форматирования надписи



- Для перехода в нижний колонтитул нажмите кнопку вниз ↓ или нажмите кнопку Верхний/Нижний колонтитул панели инструментов Колонтитул (рис.2.11).



СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ТАБЛИЦ

Таблица вставляется в текст для занесения табличных данных. Таблицы состоят из строк и столбцов, пересечение которых образует ячейку, в которых могут находиться любые данные, включая рисунки. Существует несколько способов создания таблицы: можно [вставить таблицу](#) в документ, указав ее размеры и формат, или [нарисовать](#) произвольную таблицу, используя панель инструментов Таблицы и границы.

Таблицу можно модифицировать, изменив ее размер, структуру или формат. Форматирование текста осуществляется обычным образом.

Строки и столбцы могут быть добавлены соответствующими командами пункта меню Таблица.

- Чтобы вставить (или удалить) строку или столбец, щелкните в ячейке, до или после которой осуществляется вставка. Затем в меню Таблица выделите команду Вставить (или Удалить рис. 2.28) и щелкните на нужном варианте. Если перед вставкой вы выделили несколько строк или столбцов, Word XP добавит (или удалит) такое же количество строк или столбцов.

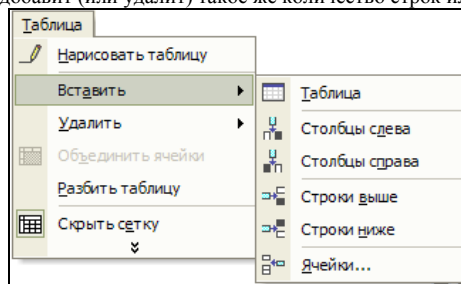



Рис.2.28. Диалоговое окно Вставка таблицы

Выделение ячеек - в левой части каждой ячейки располагается полоса выделения. При подводе к ней указателя мыши, он меняет вид. Щелчок в этой полосе выделяет ячейку. Нажатие клавиши в этой полосе и дальнейшая буксировка выделяет нужное число ячеек. Двойным щелчком можно выделить одну ячейку.


Выделение строк - щелчок (и буксировка) левой кнопкой мыши по полосе выделения строк слева от таблицы.

Выделение столбцов - щелчок (и буксировка) левой кнопкой мыши по полосе выделения столбцов выше таблицы (при этом курсор превращается в маленькую жирную стрелку).

Выделение таблицы – щелкните на кнопке Выделить таблицу над левой верхней ячейкой таблицы . Затем таблицу можно переместить в новое место.

Выделенная ячейка может быть **разбита** на нужное число ячеек по команде Таблица / Разбить ячейки.

Если данные не умещаются в ячейках или занимают слишком мало места, **измените размер** строк или столбцов перетаскиванием их границ.

Чтобы **быстро изменить размер всей таблицы**, щелкните в любой ячейке, а затем перетащите рукоятку выделения , которая появится в правом нижнем углу таблицы.

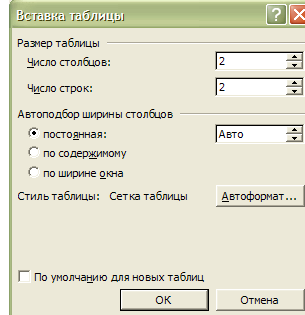
Несколько выделенных ячеек таблицы могут быть **объединены** по команде Таблица / Объединить ячейки.

Кроме того, таблица в текстовом документе применяется для разделения объектов, для которых необходимо применить разное форматирование. Сетка таблицы, показываемая бледным пунктиром, на принтере не печатается. В ячейки таблицы может быть введен текст, вставленные рисунки и другие объекты, создаваемые в Word XP. Например, с помощью таблицы размером 2 на 2 скомпонован текст и рис. 2.30.

Выполнение задания.

- Выйдите из колоннитула и перейдите в основной текст. Курсор установите строкой выше заголовка текста "ИНФОРМАЦИОННОЕ СООБЩЕНИЕ...".
- После выделения команды Таблица / Вставить / Таблица, появится диалоговое окно Вставка таблицы (рис. 2.29), в котором укажите: число столбцов – 2, число строк – 2 ⇒ **ОК**

Рис.2.29. Диалоговое окно Вставка таблицы



- Для того чтобы объединить две ячейки первой строки, **выделите всю строку** и выделите команду Таблица / Объединить ячейки.
- В первой строке введите свою фамилию (**выравнивание по центру** - рис.2.8). В первой ячейке второй строки введите текущую дату (**выравнивание по левому краю** - рис.2.8), а во вторую ячейку – текущее время (**выравнивание по правому краю** - рис.2.8).
- Этот ввод можно выполнить с помощью команд Вставка / Дата и время / выбора необходимого формата даты и времени (рис. 2.30). При следующем открытии документа дата и время будут поставлены так, как они заданы на компьютере, если флажок Обновлять автоматически, был установлен.

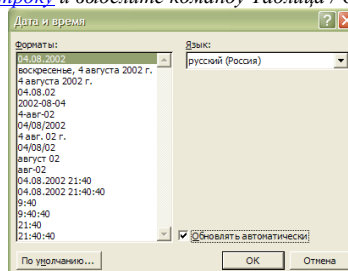


Рис.2.30. Диалоговое окно Дата и время

- Удалите ненужные границы таблицы. Для этого выделите таблицу / Формат / Границы и заливка / вкладка Граница. Сначала нажмите на клавишу **Нет границ**, а затем для верхней строки установите **Верхняя граница**, а для нижней строки – **Нижняя граница**. Результат представлен на рис. 2.31.

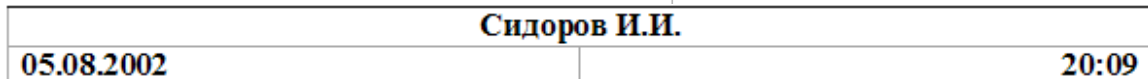
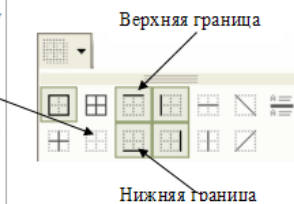


Рис.2.31. Результат редактирования таблицы

Word XP позволяет добавить в документ таблицу сложной формы путем ее рисования с помощью панели инструментов Таблицы и границы. Попробуйте нарисовать таблицу, изображенную на рис. 2.31.

Выполнение задания.

- Выберите команду Вид / Панели инструментов / Таблицы и границы (один из ее вариантов представлен на рис. 2.32).

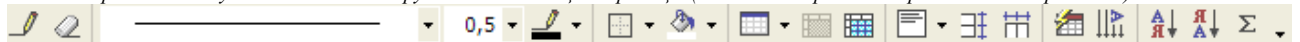




Рис.2.32. Панель инструментов Таблицы и границы

Она позволяет выполнить большое количество операций, но нас пока интересуют только те, что позволяют нарисовать таблицу сложной формы.

- Щелкните по кнопке Нарисовать таблицу . Курсор примет форму карандаша. Проводите горизонтальные и вертикальные линии до тех пор, пока не нарисуете нужную таблицу.

- Лишние границы удалите с помощью инструмента ластик, который можно вызвать, нажав на кнопку Ластик . Отключить выбранный инструмент можно повторным нажатием на его кнопку.

- Далее удалите нарисованную таблицу, так как у вас она уже есть. ВВОД ФОРМУЛ В ТЕКСТ (выполнять по указанию преподавателя)

Для написания формул предназначена программа Microsoft Equation 3.0. Она вызывается командой Вставка / Объект / Microsoft Equation 3.0. В появившуюся рамку вводится формула. Символы, присутствующие на клавиатуре, вводятся с клавиатуры. Недостающие символы выбираются из специальной панели инструментов Формула (рис. 2.33).

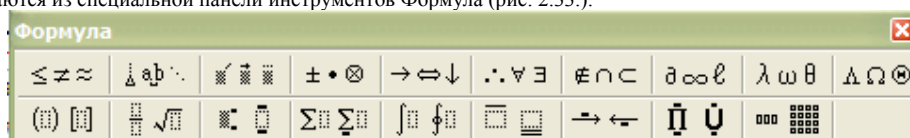


Рис.2.33. Панель инструментов Формула Microsoft Equation 3.0

- Щелчок по кнопке на панели разворачивает набор символов.
- Щелчок по символу вставляет символ в формулу.
- Щелчок вне поля ввода формулы завершает ввод.
- Для повторного редактирования формулы нужно дважды щелкнуть по ней. Вновь вызывается программа Microsoft Equation 3.0.

Выполнение задания.

Введите формулу, изображенную на рис. 2.34.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n \xi_i^2}{n(n-1)}}$$

Рис.2.34. Формула, созданная средствами Microsoft Equation 3.0

Для этого выполните следующие действия:

- Для вызова редактора формул поместить курсор в то место, где предполагаете расположить формулу и выполните команду Вставка / Объект.
- В окне диалога Вставка объекта выберите тип объекта Microsoft Equation 3.0, нажмите **ОК**.
- Установите курсор мыши на пиктограмме с греческими буквами и выберите необходимый символ.



- С помощью клавиатуры введите знак =
- Выберите на панели шаблонов кнопку



и в открывшемся списке шаблонов

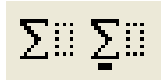
выберите



- Выберите на панели шаблонов кнопку и в открывшемся списке шаблонов выберите



- Выберите и в открывшемся окне выберите подходящий шаблон



- С помощью шаблона греческих букв введите ξ
- Далее выберите шаблон верхних и нижних подходящих



символов и вставьте.

- Установите курсор мыши в окно нижнего индекса и введите с клавиатуры i , установите курсор мыши в окно верхнего индекса и введите 2 .
- Также введите с клавиатуры параметры суммы n и $i=1$.
- Переместите курсор в знаменатель и введите $n(n-1)$.
- Измените размер формулы, если в этом есть необходимость.

ПРЕДВАРИТЕЛЬНЫЙ ПРОСМОТР И ПЕЧАТЬ ДОКУМЕНТОВ

Прежде, чем напечатать документ, следует посмотреть, как он будет выглядеть в напечатанном виде.

- Для этого откройте окно предварительного просмотра, выделив команду Файл / Предварительный просмотр или просто щелкнув на кнопке Предварительный просмотр панели инструментов Стандартная.

Документ отобразится в том виде, в котором он будет напечатан, включая колонтитулы (рис. 2.35).

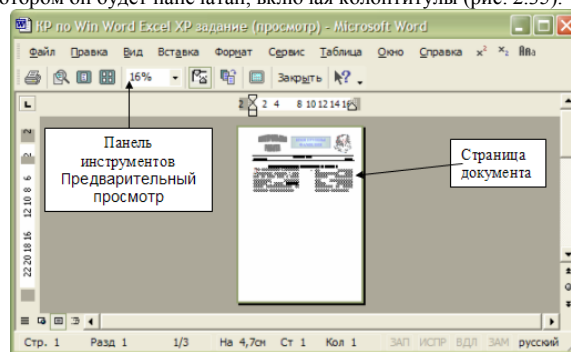


Рис.2.35. Экран предварительного просмотра документа

Панель инструментов Предварительный просмотр содержит кнопки, позволяющие отображать нужное число страниц, переключаться между страницами и изменять масштаб отображения.

Для того чтобы напечатать документ, достаточно щелкнуть по кнопке Печать на панели инструментов Стандартная или Предварительный просмотр. Документ будет напечатан на заданном по умолчанию принтере с использованием заданных по умолчанию параметров.

Для изменения параметров печати выберите команду Файл / Печать. В появившемся диалоговом окне Печать (рис. 2.36) можно выбрать другой принтер, указать число копий документа или диапазон страниц, которые нужно напечатать

Тема 2.3. Табличный процессор (лекция – дискуссия 6 часов)

Функции табличного процессора

1. Табличный процессор Quattro Pro.
2. Табличный процессор ThinkFree Calc.
3. Табличный процессор SuperCalc.
4. Табличный процессор Excel.
5. Табличный процессор abs.

Табличный процессор — это комплекс взаимосвязанных программ, предназначенный для обработки электронных таблиц.

Электронная таблица — это компьютерный эквивалент обычной таблицы, состоящей из строк и граф, на пересечении которых располагаются клетки, в которых содержится числовая информация, формулы или текст.

Табличный процессор обеспечивает работу с большими таблицами чисел. При работе с табличным процессором на экран выводится прямоугольная таблица, в клетках которой могут находиться числа, пояснительные тексты и формулы для расчета значений в клетке по имеющимся данным. То есть программные средства для проектирования электронных таблиц называют табличными процессорами. Они позволяют не только создавать таблицы, но и автоматизировать обработку табличных данных.

С помощью электронных таблиц можно выполнять различные экономические, бухгалтерские и инженерные расчеты, а также строить разного рода диаграммы, проводить сложный экономический анализ, моделировать и оптимизировать решение различных хозяйственных ситуаций и т.д.

Функции табличных процессоров весьма разнообразны:

- создание и редактирование электронных таблиц;
- создание многотабличных документов;
- оформление и печать электронных таблиц;
- построение диаграмм, их модификация и решение экономических задач графическими методами;
- создание многотабличных документов, объединенных формулами;
- работа с электронными таблицами как с базами данных: сортировка таблиц,
- выборка данных по запросам;
- создание итоговых и сводных таблиц;
- использование при построении таблиц информации из внешних баз данных;
- создание слайд-шоу;
- решение оптимизационных задач;
- решение экономических задач типа “что – если” путем подбора параметров;
- разработка макрокоманд, настройка среды под потребности пользователя и т.д.

Наиболее популярными электронными таблицами для персональных компьютеров являются табличные процессоры Microsoft Excel, Lotus 1-2-3, Quattro Pro и SuperCalc. И если после своего появления в 1982 году Lotus 1-2-3 был фактически эталоном для разработчиков электронных таблиц, то в настоящее время он утратил свои лидирующие позиции. Результаты тестирования продемонстрировали явное преимущество Excel по многим параметрам. Единственное превосходство Lotus 1-2-3 – это скорость работы, но опять же, превышение небольшое. Перспективные направления в разработке электронных таблиц основными фирмами-разработчиками определены по-разному. Фирма Microsoft уделяет особое внимание совершенствованию набора функциональных средств Excel, и в этом ее пакет явно лидирует среди всех электронных таблиц. Фирма Lotus основными усилия сконцентрировала на разработке инструментов групповой работы. Пакет Quattro Pro в результате тестирования получил достаточно высокие оценки, но ни одна из особенностей пакета не вызвала к себе повышенного внимания. Наиболее привлекательными оказались лишь возможности сортировки данных. Ситуация, сложившаяся на рынке электронных таблиц, в настоящее время характеризуется явным лидирующим положением фирмы Microsoft – 80% всех пользователей электронных таблиц предпочитают Excel. На втором месте по объему продаж – Lotus 1-2-3, затем Quattro Pro. Доля других электронных таблиц, например SuperCalc, совершенно незначительна.

Табличный процессор Quattro Pro

Среди имеющихся на рынке электронных таблиц для DOS программа Quattro Pro лучшая. Пакет Quattro Pro рассчитан практически на любую вычислительную систему – от машин с процессором 8088 и емкостью памяти 512 Кбайт до IBM PC 486 с большой оперативной памятью.

В новой версии в верхней части экрана находится программируемое “быстрое меню”, один из пунктов которого позволяет переключаться между графическим и текстовым режимами. Табличный процессор Quattro Pro обладает рядом достоинств:

- удобный пользовательский интерфейс, дающий возможность предоставления данных в самой нестандартной форме; многооконный режим работы;
- доступ к любым неограниченным по размерам внешним базам данных созданных на основе наиболее популярных СУБД;
- хорошее качество печати входных документов; легкость создания программы обработки информации в таблицах, удобные средства отладки и редактирования созданных программ и т.д.

Одной из отличительных особенностей процессора Quattro Pro являются аналитические графики, которые позволяют применять к исходным данным агрегирование, вычислять скользящее среднее и проводить регрессионный анализ; результаты перечисленных действий отражаются на графиках. Набор встроенных функций в пакете Quattro Pro включает в себя все стандартные функции. Новыми для данной версии является поддержка дополнительных библиотек @-функций, разработанных независимыми поставщиками. Данный пакет включает программы линейного и нелинейного программирования. Оптимизационную модель можно записать на рабочий лист и работать с ней. Кроме обычных команд работы с базами данных, Quattro Pro умеет читать внешние Базы в форматах Paradox, dBase и Reflex, и искать в них нужную информацию.

Табличный процессор ThinkFree Calc

Табличный процессор Calc предоставляет пользователю мощный набор инструментов для работы с табличными данными. Как и большинство современных электронных таблиц, Calc имеет формат рабочих книг 255x255x65536, т.е. основной документ может содержать 255 рабочих листов размером 255 столбцов на 65536 строк.

К ячейкам таблиц применимы стандартные для табличных процессоров операции форматирования: задание типа значения, выравнивание по горизонтали/вертикали, настройка шрифта/заливки и другие. Для удобства работы с табличными данными в программе реализованы перетаскивание содержимого ячеек, автозаполнение (копирование и построение прогрессии), скрытие/отображение строк и столбцов, закрепление областей, сортировка строк по различным критериям и прочее.

Для проведения сложных расчетов программа поддерживает несколько сотен функций, организованных в категории: математические, статистические, логические, для работы с базами данных и т.д. Функции можно вводить с помощью *специального мастера*, который, однако, значительно уступает своему аналогу из Microsoft Excel.

Функции и форматирование текста — это хорошо, но какой же табличный процессор без совершенных графических возможностей? ThinkFree Calc дает возможность наглядно представить сложные числовые зависимости с помощью графиков десятков различных типов. Основные *типы графиков* — столбиковые, круговые, линейные, точечные, поверхностные и другие, хорошо знакомые по другим программам. Кроме графиков, в документы можно вставлять внешние графические рисунки, в том числе и из клипарта.

Рабочие книги ThinkFree Calc хранятся в файлах с расширением *CLF*, поддерживается также формат *Excel* и экспорт в *HTML*.

Табличный процессор SuperCalc

SuperCalc – это один из пакетов прикладных программ. Основное применение SuperCalc – выполнение расчетов. Однако в силу своей гибкости он позволяет решать большинство финансовых и административных задач:

SuperCalc выполняет арифметические, статистические, логические, специальные функции. Он имеет дополнительные возможности: поиск и сортировка в таблицах. SuperCalc имеет довольно большие графические возможности, позволяя строить на экране семь видов диаграмм и графиков, облегчая тем самым труд пользователя. Таблицы SuperCalc могут иметь до 9999 строк и до 127 столбцов. Строки идентифицируются числами от 1 до 9999, а столбцы буквами от A до DW. По умолчанию в памяти ЭВМ резервируется место для 2000 строк и 127 столбцов. Ширина каждого столбца по умолчанию устанавливается равной 9 печатным позициям, но можно установить любую ширину столбца, внося специальную команду. На экране существует активная клетка, которая всегда подсвечивается. Такая подсветка называется табличным курсором, который можно перемещать с помощью клавиатуры.

На экране дисплея в каждый момент можно наблюдать только 20 строк таблицы и 8 столбцов стандартной ширины. Под двадцатой строкой размещаются так называемые служебные строки:

1 – строка состояния (STATUS LINE), где автоматически отображаются содержимое, координаты активной клетки и др.

2 – строка подсказки (PROMPT LINE), где высвечивается информация об ошибках

3 – строка ввода (ENTRY LINE), где высвечиваются символы, набираемые на клавиатуре.

4 – строка помощи (HELP LINE). Она показывает назначение отдельных клавиш на клавиатуре ЭВМ.

При работе в SuperCalc выделяют три основных режима работы:

Режим электронной таблицы. Здесь активен только табличный курсор. Редактирующий курсор неподвижен и находится в исходной позиции строки ввода.

Режим ввода. Он устанавливается автоматически, с началом работы на клавиатуре.

Командный режим устанавливается несколькими способами. Наиболее распространенный способ – перед набором команды нажимается команда с символом “\”.

Табличный процессор Excel

MS Excel – одна из самых популярных сегодня программ электронных таблиц. Ею пользуются ученые и деловые люди бухгалтеры и журналисты, с ее помощью ведут разнообразные таблицы, списки и каталоги, составляют финансовые и статистические отчеты, подсчитывают состояние торгового предприятия, обрабатывают результаты научного эксперимента, ведут учет, готовят презентационные материалы. Возможности Excel очень высоки.

Обработка текста, управление базами данных – программа настолько мощна, что во многих случаях превосходит специализированные программы-редакторы или программы баз данных. Такое многообразие функций может сначала запутать, чем заставить применять на практике. Но по мере приобретения опыта начинаешь по достоинству ценить то, что границ возможностей Excel тяжело достичь. За многолетнюю историю табличных расчетов с применением персональных компьютеров требования пользователей к подобным программам существенно изменились.

В начале основной акцент в такой программе, как, например, VisiCalc, ставился на счетные функции. Сегодня наряду с инженерными и бухгалтерскими расчетами организация и графическое изображение данных приобретают все возрастающее значение. Кроме того, многообразие функций, предлагаемое такой расчетной и графической программой, не должно осложнять работу пользователя. Программы для Windows создают для этого идеальные предпосылки. В последнее время многие как раз перешли на использование Windows в качестве своей пользовательской среды. Как следствие, многие фирмы, создающие программное обеспечение, начали предлагать большое количество программ под Windows.

Программа Excel обеспечивает как легкость при обращении с данными, так и их сохранность. Excel позволяет быстро выполнить работу, для которой не нужно затрачивать много бумаги и времени, а также привлекать профессиональных бухгалтеров и финансистов. Данная программа умеет вычислить суммы по строкам и столбцам таблиц, посчитать среднее арифметическое, банковский процент или дисперсию, здесь вообще можно использовать множество стандартных функций: финансовых, математических, логических, статистических.

Так как Excel – программа, работающая с таблицами, то здесь оперируют различными данными. В электронных таблицах используют, как правило, следующие типы данных: текст – это любая последовательность символов; число – это числовая константа; формула – это выражение, состоящее из числовых величин и арифметических операций (Пример: =A5/H8*12); функции – это запрограммированные формулы, позволяющие проводить часто встречающиеся последовательности вычислений (Например, функция автосуммирования может быть представлена следующим образом: =СУММ(A1: A4)).

Оформление таблиц может быть самым разнообразным, возможности форматирования данных – как в хорошем текстовом процессоре: можно менять шрифты, начертания, выделять строки, столбцы или отдельные ячейки текста цветом, рамочками и линеечками, закрашивать области фоном или цветом, строить по табличным данным графики и диаграммы, вставлять таблицу с картинками и т.д.

Программа достаточно мощная, возможности ее, особенно в последних версиях, весьма обширны. Одних только математических, логических, бухгалтерских, статистических функций, которые Excel может выполнять над табличными данными более 200.

Excel – программа многооконная, что позволяет нам одновременно загружать столько файлов, сколько позволит объем оперативной памяти компьютера. Окно Excel содержит множество различных элементов. Некоторые из них присущи всем программам в среде Windows, остальные есть только в окне Excel. Вся рабочая область окна Excel занята чистым рабочим листом (или таблицей), разделенным на отдельные ячейки. Столбцы озаглавлены буквами, строки – цифрами. Как и во многих других программах в среде Windows, вы можете представить рабочий лист в виде отдельного окна со своим собственным заголовком – это окно называется окном рабочей книги, так как в таком окне можно обрабатывать несколько рабочих листов. На одной рабочей странице в распоряжении будет 256 столбцов и 16384 строки. Строки пронумерованы от 1 до 16384, столбцы названы буквами и комбинациями букв. После 26 букв алфавита колонки следуют комбинации букв AA, AB и т.д.

После запуска Excel содержит пять областей: окно книги, которое занимает большую часть экрана, строку меню, две или больше панелей инструментов, строку формул и строку состояния (строка меню, панелей инструментов, строка формул, и строка состояния появляются на экране даже в том случае, если книга не видна). Все вместе эти пять областей называются «Рабочей областью Excel». После запуска Excel первой пока еще пустой книги дается имя Книга 1. Если в течении текущего сеанса работы будет открыта новая книга, то Excel назовет ее Книга 2. Книга Excel может содержать листы пяти типов: рабочие листы, листы диаграмм модули Visual Basic, листы диалогов и листы макросов Microsoft Excel.

Окно книги составляет основную часть рабочей области. В нижней части окна книги размещаются кнопки прокрутки ярлыков листов, а в верхней части – строка заголовка. Новая книга первоначально содержит 16 отдельных листов. Чтобы просмотреть содержание книги, можно использовать четыре кнопки, расположенные в нижнем левом углу окна. Две средние кнопки служат для прокрутки на один лист влево или вправо. Две крайние кнопки выполняют прокрутку к первому или последнему листу книги.

Перечисленные кнопки прокрутки не активизируют листы книги. Чтобы сделать лист активным, следует после прокрутки ярлыков щелкнуть на листе. В правом конце строки заголовка окна книги находятся три кнопки для управления размерами окон: Свернуть, Развернуть и Заккрыть. Ячейка, находящаяся на пересечении строки и столбца, является основным элементом любого рабочего листа. Каждая ячейка занимает уникальное место на листе, может хранить и отображать информацию, имеет однозначные координаты, которые называются адресом ячейки или ссылкой. Например, ячейка, находящаяся на пересечении столбца A и строки 1, имеет адрес A1. Ячейка на пересечении столбца Z и строки 100 имеет адрес Z100. Ссылки, являющиеся идентификаторами ячеек, бывают трех видов: абсолютные, относительные и смешанные. Абсолютные не изменяются, когда ячейки содержат формулы при копировании.

В относительных ссылках адреса при копировании формул в другое место изменяются. Смешанные ссылки состоят из абсолютных и относительных. В тех случаях, когда необходимо, чтобы изменились координаты ячеек используют относительные ссылки, если необходимо, чтобы координаты не изменились используют относительную ссылку, в иных случаях используют смешанные. В тех случаях, когда координаты следует делать, неизменными перед ними ставят знак “\$”. Выделенную ячейку называют активной или текущей ячейкой, адрес активной ячейки выводится в поле имени, которое находится в левом конце строки формул. При 256 столбцах и 16384 строках рабочий лист содержит более 4 миллионов ячеек.

Макрос - это записанная последовательность команд и действий пользователя, сохранённая под уникальным именем, которую может выполнить Excel.

Код макроса может служить основой для дальнейших разработок и является учебным материалом, т.к. по коду макроса можно научиться записывать последовательность действий в VBA.

Имя макроса может содержать до 255 символов и должно начинаться с буквы. В имя могут входить буквы, цифры и знаки подчёркивания. Пробелы в именах макросов не допускаются.

Если при работе с Microsoft Excel для Windows возникает необходимость несколько раз выполнить одну и ту же последовательность действий, то можно записать эту последовательность под определенным именем. Записанная под определенным именем последовательность действий называется макросом. Записанный макрос можно вызывать для выполнения из основного меню при помощи кнопки на панели инструментов или на рабочей области, а также комбинации клавиш. Макрос может быть назначен так же графическому объекту.

Обозначим основные моменты создания макроса.

1. Макрос представляет собой записанную последовательность определенных действий.

2. Для записи макроса выполните команду Сервис | Запись макроса.

3. Назначение существующему макросу клавиши быстрого вызова и пункта меню выполняется с помощью команды Сервис | Макрос.

4. Команда С относительными ссылками записывает макрос, обрабатывающий данные в любом месте рабочей книги.

5. Редактировать текст макроса можно непосредственно в модуле с текстом.

6. Для удаления макроса нажмите мышью на его имени. При этом станут доступными кнопки окна диалога “Макрос”, в том числе кнопка Удалить.
7. Команда Сервис | Назначить макрос назначает макросы графическим объектам.
8. Введите в поле ввода Имя макроса название создаваемого макроса и запишите новый макрос.
9. Вы можете вносить изменения непосредственно в текст программы, в которой на языке Visual Basic for Applications записана последовательность действий макроса.
10. Записанный макрос можно запустить на выполнение одним из следующих способов:
 - из основного меню
 - при помощи кнопки на панели инструментов
 - при помощи кнопки на рабочей области
 - нажатием комбинации клавиш клавиатуры
 - нажатием кнопкой мыши на графическом объекте.

Табличный процессор abs

Табличный процессор abs начинал свою историю как попытка автора заработать деньги. Начиная с версии 0.4 (текущая 0.6) он распространяется по GNU General Public License.

Поэтому, исходные тексты доступны и проблема русификации разрешима.

Сам по себе abs поражает своей компактностью - объем исходных текстов всего 263К, а между тем в них влезает полный интерпретатор Бейсик.

По своим возможностям он более-менее сопоставим со многими современными таблицами - это скорее инструмент для рисования и оформления, чем мощная расчетная система, каковыми были Lotus 1-2-3 и Quattro Pro.

Впрочем, большинство современных офисных пользователей не нуждаются в матричной алгебре, а основные возможности табличного процессора - формулы, суммирование, простейшие графики в abs есть. А еще есть встроенный бейсик, на котором систему можно заточить под конкретное применение. Конечно же, любой продвинутый пользователь Linux недовольно сморщит нос, услышав о необходимости писать на бейсике...

Несколько неудобным является использование специальных команд Copy Right и Copy Down для размножения формул вместо просто Copy или механизма Cut&Paste как в Excel. Но, видимо, к этому можно привыкнуть.

Теперь о русификации. Сам табличный процессор русифицируется очень просто - в нем уже есть поддержка Xkb, и единственным камнем преткновения являются прошитые в код имена шрифтов - adobe-times, adobe-courier и -adobe-helvetica. Взяв исходники abs и заменив в файле font.c adobe на stopux, получаем результат - abs начинает разговаривать по-русски. Остается печатать.

Печать в abs реализована следующим образом: записывается файл в формате fig и пропускается через transfig для получения постскриптовского файла. Решение замечательное своим изяществом, но, к сожалению, возвращает нас к проблеме русификации transfig (а хорошо бы еще и xfig - для предварительного просмотра). Впрочем, это не обязательно. Можно просто менять имена шрифтов в выходном постскриптовском файле.

Табличный процессор Lotus 1-2-3

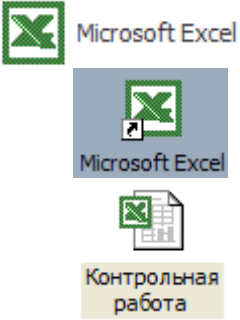

Lotus 1—2—3 — профессиональный процессор электронных таблиц. Широкие графические возможности и удобный интерфейс пакета позволяют быстро ориентироваться в нём. С его помощью можно создать любой финансовый документ, отчет для бухгалтерии, составить бюджет, а затем разместить все эти документы в базах данных.

Табличный процессор

Вы вновь вернулись в окно своей группы. Теперь вам предстоит ознакомиться с основами работы с *Microsoft Excel XP*, предназначенного для организации данных в таблицы для документирования, сопоставления и графического представления информации.

ЗАГРУЗКА EXCEL XP

☛ Выполнение задания

- В окне группы щелкните правой кнопкой мыши. Затем [Создать / Лист Microsoft Excel](#) (рис.1.28) – этот вариант предпочтительный, так как он сразу создает документ в нужном месте или
 - Нажмите кнопку [ПУСК](#) на Панели задач / Все программы и выберите левой кнопкой мыши команду Microsoft Excel или
 - Нажмите кнопку [ПУСК](#) на Панели задач / в левом столбце щелкните по значку Microsoft Excel
- 
- Дважды щелкните по ярлыку запуска Excel XP или
 - Дважды щелкните по значку ранее созданного пустого документа Microsoft Word XP, или
 - Чтобы открыть документ, с которым вы недавно работали, нажмите кнопку [Пуск](#)
- 

остановите указатель на пункте Недавние документы и щелкните мышью по документу, который желательно запустить.

НАСТРОЙКА ЭКРАНА

При запуске Excel XP появляется пустой документ (рис. 3.1).

На рис 3.1. показаны наиболее важные элементы окна. Нововведением Excel XP является Область задач, и диалоговое окно Введите вопрос.

- Диалоговое окно Введите вопрос, позволяет ввести вопрос и Excel XP выдаст помощь по соответствующей теме.
- Область задач (аналогична области [Создание документа](#) в Word XP рис. 2.3), позволяет открывать файлы, вставлять информацию из буфера обмена и создавать как новые, так и базирующиеся на существующих файлах Рабочей книги. Область задач объединяет множество общих задач и позволяет выполнять их одним щелчком мыши.

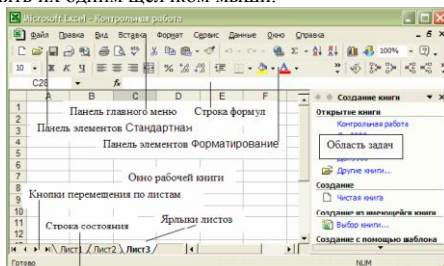


Рис.3.1. Внешний вид открытого листа Excel XP

Область задач можно вызывать и убирать командой Вид / Область задач. Ее можно также убрать, нажав на кнопку с изображением крестика в верхней части области.

☞ **Выполнение задания**

- Щелкните по кнопке **Развернуть** в зоне заголовка окна, если после вызова программы ее окно не занимает всего экрана. Проверьте, чтобы полноэкранным было и окно документа.



- Выберите команду **Вид / Панели инструментов** и установите флажки перед панелями инструментов **Стандартная** и **Форматирование**, убрав все остальные.

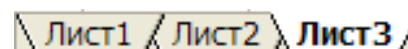
СОХРАНЕНИЕ ДАННЫХ И ЗАВЕРШЕНИЕ РАБОТЫ С EXCEL XP

Сохранение данных и завершение работы с Excel XP выполняется аналогично, как и в Word XP. Первый раз можно выполнить команду **Сохранить как**.

☞ **Примечание 1.** В связи с неустойчивой работой компьютеров, рекомендуем периодически выполнять команду **Сохранить**. Этим Вы сэкономите свой труд.

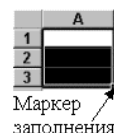
ОСНОВНЫЕ ПОНЯТИЯ ЭЛЕКТРОННЫХ ТАБЛИЦ EXCEL XP

- Рабочее поле** ЭТ состоит из строк и столбцов. Максимальное количество строк равно - 65536, столбцов - 256. Пересечение строки и столбца образует **ячейку**, в которую можно вводить данные (текст, число, дату и время или формулы).
- Строки** нумеруются цифрами на левой границе рабочего поля.
- Столбцы** (колонки) нумеруются латинскими буквами на верхней границе рабочего поля. Колонки нумеруются в следующем порядке: A...Z, затем AA...AZ, затем BA...BZ и т. д. до IV.
- Каждая ячейка имеет уникальный **адрес**, состоящий из букв столбца и номера строки. Например, адрес C7 определяет ячейку на пересечении столбца C и строки 7.
- Указатель ячейки** - светящийся прямоугольник, определяет **текущую ячейку**. Указатель можно перемещать по таблице мышью или с помощью клавиатуры. Ввод данных и некоторые другие действия по умолчанию относятся к текущей ячейке.
- Блок (область)** - прямоугольная область смежных ячеек. Блок может состоять из одной или нескольких ячеек, строк или столбцов. Адрес блока состоит из координат противоположных углов, разделенных двоеточием. Например, A1:C7 - блок из трех столбцов и семи строк, D:F - блок из всех ячеек столбцов D, E и F.
- Рабочий лист, книга.** Окно с заголовком Книга состоит из нескольких рабочих листов (по умолчанию - 16, но может быть установлено любое количество). При открытии рабочей книги автоматически загружаются все ее рабочие листы. На экране виден только верхний лист, остальные представлены ярлычками. Щелкая мышью на ярлычках листов, можно перейти к другому листу.



ВЫДЕЛЕНИЕ СТОЛБЦОВ, СТРОК, БЛОКОВ, ТАБЛИЦЫ

- Столбец** - щелкните мышью на имени столбца.
- Несколько столбцов** - не отпуская кнопку после щелчка, протяните мышью по именам столбцов.
- Строку** - щелкните мышью на имени строки
- Несколько строк** - не отпуская кнопку после щелчка, протяните мышью по именам строк.
 - Блок** - щелкните мышью на начальной (конечной) ячейке блока и, не отпуская кнопку, протяните курсор мыши на конечную (начальную) ячейку. В любом случае адрес блока будет Верхняя левая ячейка...; Правая нижняя ячейка. Первая выделенная ячейка диапазона остается белой.



Она является активной и для нее вы можете писать **формулы массива**.

В правом нижнем углу выделенной ячейки (блока ячеек) появляется черный квадрат, именуемый маркером заполнения. Он используется для создания **последовательности** (п.3.6.8), или **"копирования" формул**.

Для снятия выделения достаточно щелкнуть мышью по любому невыделенному участку рабочего листа. Новое выделение заменяет предыдущее.

Чтобы выделить диапазон ячеек с помощью клавиатуры:

- Поместите указатель ячейки на первую ячейку **выделяемого** диапазона.
- Нажмите клавишу **Shift** и, удерживая ее нажатой, расширьте выделяемый диапазон в нужном направлении с помощью клавиш управления курсором.

Чтобы выделить диапазон несмежных ячеек:

- Выделите первый диапазон ячеек.
- Удерживайте клавишу **Ctrl** нажатой и выделяйте мышкой следующие диапазоны.

Предыдущие диапазоны останутся выделенными.

ВВОД И ИЗМЕНЕНИЕ ДАННЫХ В РАБОЧЕМ ЛИСТЕ

Чтобы поместить данные в ячейку рабочего листа, следует выделить нужную ячейку (блок ячеек), ввести данные и подтвердить ввод.

ВЫПОЛНЕНИЕ ВВОДА ДАННЫХ

Ввод данных в ячейку не зависит от типа вводимых данных. Excel XP сам определяет, являются ли вводимые данные текстом, числом или формулой по первому вводимому символу.

- Поместите указатель ячейки на нужной ячейке (блоке ячеек).
- В строке состояния должно отражаться сообщение **Готово**. В поле имени левой части строки формул будет указан адрес активной ячейки.
- Подтвердите ввод данных в ячейку:
 - Нажмите клавишу **Enter** для ячейки (комбинацию клавиш **Ctrl + Enter** - для блока ячеек) или
 - Нажмите любую клавишу управления курсором или
 - Выделите любую другую ячейку листа или
 - Щелкните мышкой на кнопке **Enter** в строке формул



☞ **Примечание 1.** Для отмены **текущего ввода** данных в ячейку нажмите клавишу **Esc**, или щелкните мышкой на кнопке **Отмена** в строке формул.

☞ **Примечание 2.** Для отмены ввода данных в ячейку **после подтверждения** данной процедуры выберите команду **Правка / Отменить** изменения.

☞ **Примечание 3.** Ввод **формулы** в ячейку завершается только нажатием клавиши **Enter** или щелчком мыши на кнопке **Изменить формулу** строки формул.

☞ **Примечание 4.** Вначале вводите данные в ячейки, а затем выполняйте редактирование их внешнего вида.

ВВОД ЧИСЕЛ

- Excel интерпретирует содержимое ячейки как число, если оно состоит только из цифр. При вводе можно также использовать некоторые специальные символы: "-" - минус, "," - запятая, "." - точка, "/" - знак деления. Числовые значения по умолчанию выровнены по правому краю.
- Если ширина введенного числа превышает ширину ячейки, то Excel изображает его в экспоненциальной форме, либо вместо числа ставит символы "####" (при этом число в ячейке будет сохранено полностью). Чтобы отобразить число, увеличьте ширину столбца.
- Если вводимое числовое значение должно быть интерпретировано в качестве текста, то перед числом поставьте символ "апостроф" (одновременное нажатие клавиш **Shift + Э** в английской раскладке клавиатуры). **Ввод неправильного десятичного разделителя (точ-**

ки вместо запятой) превращает его в текст.

☞ **Примечание.** Рекомендуем вводить числа с цифровой клавиатуры. Клавиша, на которой изображена точка, и слово **Del** покажет Вам, какой разделитель используется для отделения дробной части. Обычно, это "запятая", но, иногда, при настройке Windows XP, в качестве разделителя может быть установлен знак "десятичная точка".

ВВОД ЗНАЧЕНИЙ ДАТЫ И ВРЕМЕНИ

- В ячейки рабочего листа можно ввести значения даты в интервале от 01.01.1900 до 31.12.2078. Для разделения отдельных частей значения даты наряду с точкой можно использовать косую черту "/" или дефис "-".
- Введенное время разделяется двоеточием, например, 13:21:34.
- Excel обрабатывает значения даты и времени как десятичные числа. Значение даты сохраняется от 1 (01.01.1900) до 65380 (31.12.2078). Значение времени находится в виде десятичной дроби в интервале от 0 до 1.
- Значения даты и времени по умолчанию выравниваются по правому краю.

ВВОД ФОРМУЛ

- В виде формулы может быть записано арифметическое выражение - последовательность чисел или ссылок на ячейки, объединенных знаками арифметических операций или функциями.
- Формула должна начинаться со знака "равно". Она может содержать до 256 символов, включая пробелы.
- По умолчанию, после фиксации формулы в ячейке отображается результат вычислений по заданной формуле.
- Формула вводится в ячейку и одновременно в строку формул, расположенную в верхней части листа рабочей книги Excel XP. Дальнейшее редактирование формулы возможно в обоих местах.

Операторы в формулах

Операторы определяют способ вычисления результата (значения) на основании отдельных элементов **формулы**. Excel различает три вида операторов:

1. **Арифметические** операторы "+, -" - Сложение и Вычитание, "*", "/" - Умножение и Деление, "%" - Определение значения процента, "^" - Возведение в степень
2. Операторы **сравнения**: **ИСТИНА** или **ЛОЖЬ**, "=", "<" - Больше, ">" - Больше или равно, "<=" - Больше или равно, ">=" - Больше или равно, "<>" - Не равно
3. **Текстовый** оператор
& Объединяет отдельные фрагменты текста.

Чтобы написать формулу:

- Поместите указатель ячейки на ячейку, в которую вы хотите ввести формулу.
- Укажите в качестве первого символа знак **равенства** (клавишей "равно" или кнопкой **Изменить формулу** в строке формул).

=2+2

- Введите часть формулы вплоть до позиции первой **ссылки**. Формулу следует вводить с клавиатуры как обычный текст, в **английской** раскладке клавиатуры.
- Задайте ссылку на нужную ячейку или на диапазон ячеек. Ссылки могут быть записаны как *прописными*, так и *строчными* буквами. Если ссылка задана правильно, то после подтверждения ввода формулы Excel XP преобразует все буквы в *прописные*.
- Введите оставшуюся часть формулы. Завершите ввод формулы нажатием клавиши **Enter**.

В ячейке представлен **результат вычисления формулы**. Саму формулу можно теперь увидеть в строке формул. Этот способ, однако, нередко *требует больших затрат времени и к тому же, сопряжен с ошибками*.

Чтобы задать ссылки на ячейку методом указания:

- Введите часть формулы вплоть до того места, в котором должна быть указана ссылка.
 - Поместите указатель ячейки на ячейку, ссылка на которую должна быть указана в формуле, щелкнув по ней мышкой.
- В формуле в ячейке и в строке формул представлена относительная ссылка на выделенную ячейку
- Введите следующий оператор для задания диапазона ячеек или продолжите ввод оставшейся части формулы.
 - По мере необходимости при вводе оставшейся части формулы задавайте ссылки на ячейки/диапазоны описанным способом. Завершите ввод формулы нажатием клавиши **Enter** или щелчком мыши на кнопке Изменить формулу.

☞ **Примечание.** Нажатием клавиши **F4** можно преобразовать относительную ссылку на ячейку в абсолютную.

3.6.5. ВВОД ТЕКСТА

- Любые вводимые данные, которая программа не воспринимает в качестве **числового значения** (п.3.6.2), **даты, времени** (п.3.6.3), или в качестве формулы, интерпретируются программой как текст.
- Любые данные, перед которым при вводе указан **апостроф** (п.3.6.2), воспринимаются как текст.
- Длина текста в одной ячейке - до 255 символов.
- По умолчанию текст выравнивается по левому краю.
- Если введенный в ячейку текст полностью не помещается в одной ячейке, то он будет отображен поверх расположенной справа **пустой** ячейки. Если правая ячейка не пуста, то текст на экране будет обрезан по правому краю ячейки.

C	D	E
ЛЕВАЯ ЯЧЕЙКА		ПРАВАЯ ЯЧЕЙКА

ОТМЕНА И ПОВТОРЕНИЕ ВВОДА ДАННЫХ И КОМАНД

- Последнюю команду можно отменить нажатием на кнопку Отменить на панели Стандартная. Рядом с этой кнопкой расположена кнопка списка. При нажатии на нее появляется список последних команд. При выборе команды отменяются все команды вплоть до выбранной.
- Повторение последней (нескольких последних команд) возможно при нажатии на кнопку Вернуть панели Стандартная.



3.6.7. РЕДАКТИРОВАНИЕ СОДЕРЖИМОГО ЯЧЕЕК

- Редактировать числа, даты, формулы, текст можно активизировав ячейку двойным щелчком левой кнопки мыши или нажав на клавишу **F2**.
- Содержимое ячейки также можно редактировать в Строке формул, где дублируется содержимое **активной** ячейки. Курсор устанавливается в нужное место Строки формул и редактируется содержание ячейки.
- Окончить редактирование, можно **только** нажав на клавишу **Enter** или щелкнув мышкой на кнопке **Enter** в строке формул



☞ **Примечание.** Для **полного изменения** данных, находящихся в ячейке, достаточно **вновь** ввести в нее новые данные. Предыдущее содержимое будет удалено.

Выполнение задания.

На первом этапе Вам надо будет создать таблицу следующего вида (рис. 3.2.):

	A	B	C	D	E	F
	Магазины	Товарооборот план, тыс.руб	Товарооборот факт, тыс.руб	% выполнения	Доля	Премия, тыс. руб
1						
2	1	321	462	?	?	?
3	2	451	400	?	?	?
4	3	325	380	?	?	?
5	3	451	450	?	?	?
6	2	254	300	?	?	?
7	1	465	541	?	?	?
8	Итого	?	?	?	?	?
9	Максимум	?	?	?	?	?

В ячейку A1 введите следующий текст - Магазины. Нажмите клавишу клавиатуры **Перемещение вправо**. В ячейку B1 введите текст - Товарооборот план, тыс. руб. Вновь нажмите клавишу **Перемещение вправо**. Повторяйте эти действия для блока ячеек C1:F1. Не обращайте внимания на то, что текст в ячейках накладывается друг на друга. Немного позже, Вы его отформатируете. В ячейку C1 введите текст - Товарооборот факт, тыс. руб., D1 - % выполнения (% вводится одновременным нажатием клавиш **Shift + %** верхнего ряда клавиш), E1 - Доля, F1 - Премия, тыс.руб. Результат ввода приведен ниже (на рис. 3.3.)

	A	B	C	D	E	F	G
1	Магазины	Товарообс	Товарообс	% выполн	Доля	Премия, тыс. руб	

Рис.3.3. Не форматированный ввод данных заголовков столбцов

Пропустите ячейки A2:A7. В ячейки B2:C7 введите числа из бланка задания. В ячейку A8 введите текст - Итого, A9 - Максимум. В ячейках, в которых стоят вопросительные знаки, вам необходимо будет выполнить вычисления. **Сами вопросительные знаки вводить не надо.**

Номера магазинов (ячейки A2:A7) можно ввести, используя приемы ввода последовательностей.

ЗАПОЛНЕНИЕ ЯЧЕЕК ЗНАЧЕНИЯМИ РЯДОВ ДАННЫХ

После ввода одного или нескольких начальных значений ряда данных можно с помощью операции автозаполнения заполнить выделенные ячейки значениями ряда.

- Введите начальное значение ряда и вновь выделите ячейку. В качестве начальных значений ряда можно указывать числа, значения даты и времени, а также порядковые значения, типа "1 квартал".
- Перетащите **маркер заполнения** (черный квадратик в правом нижнем углу ячейки или блока ячеек - п.3.5) в нужном направлении. Выделенный ряд ячеек будет заполнен соответствующими значениями.
- При перемещении маркера заполнения **вниз** или **вправо** значения ряда **возрастают**.
- При перемещении маркера заполнения **вверх** или **влево** значения ряда **убывают**.

После завершения процесса автозаполнения появится кнопка **Параметры автозаполнения** (рис. 3.4). Можно выбрать один из четырех вариантов заполнения. В вашем случае можно ничего не делать.

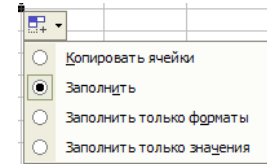


Рис.3.4. Контекстное меню кнопки Параметры автозаполнения

Примечание 1. Чтобы создать ряд данных с шагом, отличным от единицы, необходимо указать хотя бы два первых значения ряда. Выделите ячейки, содержащие эти значения, и перетащите маркер заполнения для выделенного диапазона.

Примечание 2. Если указанное значение Excel XP не воспринимает в качестве начального значения ряда (например, любое число), то либо перетащите маркер заполнения для двух первых значений ряда либо при перетаскивании маркера удерживайте нажатой клавишу **Ctrl**.

Выполнение задания.

• В ячейку A2 введите число 1. Подведите курсор мыши к маркеру заполнения. Курсор превратится в маленький крестик. Нажмите, и не отпуская, клавишу **Ctrl**. Буксируйте маркер заполнения в ячейку A4. При отпуске клавиши мыши выделенная область заполнится последовательностью чисел 1,2,3.

• В ячейки A5 и A6 введите числа 3 и 2 соответственно. Выделите область ячеек A5 и A6. Захватите и буксируйте **маркер заполнения** в ячейку A7. Так как, шаг изменения последовательности равен -1, то в ячейке A7 появится число 1.

Выполните форматирование заголовков столбцов в диапазоне ячеек A1:F1:

• Вначале выделите весь диапазон. Для этого установите курсор в ячейку A1, и, не отпуская левой кнопки мыши, переместите курсор в ячейку F1.

Результат выделения

будет иметь следующий вид (рис.3.5.):

	A	B	C	D	E	F	G
1	Магазины	Товарообс	Товарообс	% выполн	Доля	Премия, тыс. руб	

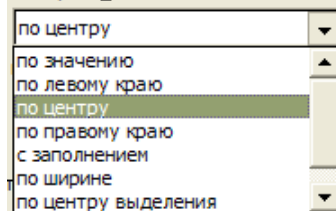
Рис.3.5. Результаты выделения данных заголовков столбцов

Первая ячейка выделяемого диапазона остается белой, а остальные будут затемнены.

- Выделите команду **Формат / Ячейки / перейдите на вкладку Выравнивание** (рис. 3.6).
- С помощью мыши установите все значения, как показано на рисунке:

Выравнивание по вертикали - по центру;
Выравнивание по горизонтали - по центру;

по горизонтали:



Отображение - переносить по словам;

Ориентация - слово НАДПИСЬ переместите из горизонтального положения в вертикальное.

- Нажмите кнопку **ОК**.

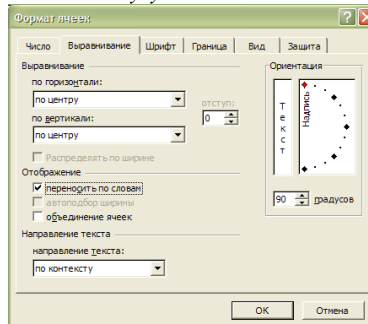
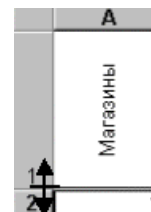


Рис. 3.6 Диалоговое окно Формат ячеек с установленными параметрами

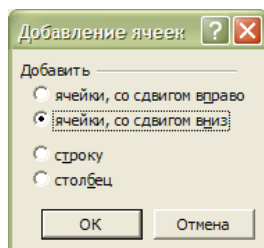
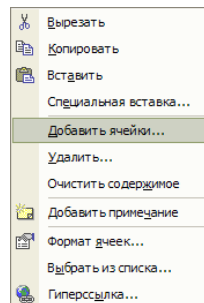
- Установите курсор мыши в заголовке строк (между первой и второй строкой) и, при нажатой левой кнопки мыши увеличьте высоту первой строки, таким образом, чтобы текст в ячейках A1:F1 стал выглядеть как [задача](#) (п.3.6.7).



ВСТАВКА, УДАЛЕНИЕ СТОЛБЦОВ, СТРОК И ЯЧЕЕК

- Чтобы вставить одну строку – [выделите строку](#) в указателе строк (несколько строк – [выделите необходимое количество](#) строк - п.3.5), затем подведите курсор к выделенному диапазону. Щелкните в его площади правой кнопкой мыши. В появившемся контекстном меню (рис. 3.7) выберите команду Добавить ячейки.
- Таблица будет разрезана вдоль, добавится новая строка (строки), существующие строки будут смещены *вниз*.
- Вставка [столбцов](#) выполняется аналогично, с учетом того, что существующие столбцы будут сдвинуты *вправо*.

Рис. 3.7. Контекстное меню редактирования и форматирования ячеек



Для вставки ячеек (блока ячеек) [выделите ячейку \(блок ячеек](#) - п.3.5). Щелкните в выделенном диапазоне правой кнопкой мыши. В появившемся контекстном меню выберите команду Добавить ячейки. В появившемся подменю (рис. 3.8.) выберите необходимый вариант сдвига существующих ячеек.

Рис. 3.8. Контекстное меню Добавления ячеек

- Удаление столбцов, строк и ячеек выполняется аналогично их вставке, за исключением того, что вместо команды Добавить ячейки выбирается команда Удалить.

☞ Выполнение задания.

Выполните добавление столбцов, строк и ячеек в таблице, а затем удалите их до исходного состояния.

3.8. ВЫПОЛНЕНИЕ НЕОБХОДИМЫХ ВЫЧИСЛЕНИЙ

☞ Выполнение задания.

Выполните суммирование в столбцах Товарооборот план, тыс. руб. одним из трех способов:

1. [Выделите блок ячеек](#) B2:B8. Щелкните по кнопке Автосумма панели инструментов Стандартная. В ячейке B8 появится значение 2267.



2. Сделайте активной ячейку B8. Щелкните по кнопке Автосумма панели инструментов Стандартная. Вокруг блока ячеек B2:B7 появится движущаяся рамка. Если Вы согласны с выделенным диапазоном, то завершите ввод формулы, в противном случае, левой кнопкой мыши [выделите нужный диапазон ячеек](#) и, [завершите ввод формулы](#).

3. Перейдите на английский и самостоятельно напишите формулу [СУММ\(B2:B7\)](#).

- Любым из способов, описанных в предыдущем пункте, подсчитайте итог по столбцу Товарооборот факт, тыс. руб. по формуле [СУММ\(C2:C7\)](#), или перетащите [маркер ячейки](#) B7 в ячейку C7. Должно получиться число 2533.

- В ячейку B9 вставьте функцию [МАКС\(B2:B7\)](#), используя кнопку, [Вставка функций](#) или [напишите ее сами](#). Должно получиться число 400.

- Сделайте активной ячейку B9. Щелкните по кнопке раскрытия списка Автосумма (рис. 3.9). Можно выбрать сразу Максимум, но попробуем найти ее в списке всех функций Excel XP. Поэтому щелкните мышью на Другие функции.

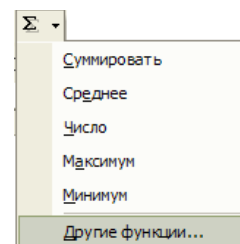


Рис. 3.9. Контекстное меню выбора функций

- В окне Категории (рис. 3.10) выберите) Статистические, а в окне Выберите функцию - МАКС

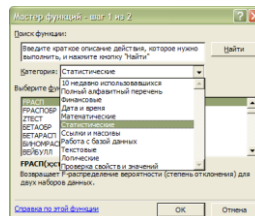



Рис. 3.10. Первое окно Мастера функций

- В окне Число 1 (рис.3.11) показано выделение области B2:B8. Но Вы с этим не согласны, поэтому нажмите на кнопку  Перехода в правой части окна и самостоятельно выделите необходимый диапазон ячеек B2:B7, или напишите его сами в английской раскладке клавиатуры.

- Вновь щелкните по кнопке Перехода для возврата во второе окно мастера функций.

- Нажмите клавишу **Enter** или щелкните по кнопке **OK**.

В ячейке появится значение 465. Аналогично выполните вычисление в ячейке C9, или с помощью [маркера заполнения](#) скопируйте формулу из ячейки B9.

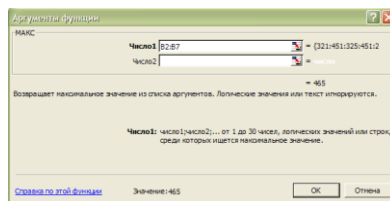


Рис. 3.11. Диалоговое окно функции расчета максимального значения Макс

• Процент выполнения плана вычисляется делением значений ячеек столбца Товарооборот факт на Товарооборот план. Установите курсор в ячейке D2. Нажмите клавишу "=", в ячейке и в строке формул появится знак "=". Щелкните левой кнопкой мыши в ячейке C2. На цифровой клавиатуре нажмите кнопку "/". Затем щелкните в ячейке B2. В результате будет написана следующая формула $=C2/B2$. Закончите ввод формулы (п.3.6.1).

• Заполните ячейки D3:D8 формулой из ячейки D2 с помощью Маркера заполнения. Установите курсор на маркере заполнения, и, не отпуская левой клавиши мыши, переместите маркер из ячейки D2 до ячейки D8. При перемещении маркера вниз будут увеличиваться номера строк в формуле.

• Эти же формулы можно написать по другому. Выделите блок ячеек D2:D8. В ячейку D2 введите формулу $=C2/B2$. Нажмите сочетание клавиш **Ctrl + Enter**. Во всех ячейках выделенного диапазона появятся нужные формулы.

¶ **Примечание.** Обратите внимание, что процент выполнения плана товарооборота считается по формуле, аналогичной для каждой группы товаров. Суммирование процентов приведет к смешному результату. Сравните, в первом случае имеем 111,73%, а во втором - 683,77%.

• Подсчитайте удельный вес каждой группы товаров каждого магазина в общем, фактическом товарообороте, как отношение фактического товарооборота к суммарному фактическому товарообороту по всем группам товаров всех магазинов:

• В ячейку E2 введите следующую формулу $=C2/C8$.

• Заполните ячейки E3:E8 формулой из ячейки E2 с помощью маркера заполнения. При перемещении маркера вниз будут увеличиваться номера строк в формуле, что приведет к появлению сообщения **#ДЕЛ/0!**. Действительно, при перемещении маркера заполнения вниз увеличивается адрес ячейки, а он во всех формулах должен быть C8. Решить эту проблему можно тремя путями:

1. Сделайте активной ячейку E2. Войдите в режим редактирования ячейки (п.3.6.7). Курсор установите после C8 (в ячейке или строке формул) и нажмите клавишу F4 столько раз, пока вместо текста C8 не появится \$C\$8. Адрес ячейки с суммарным товарооборотом стал абсолютным и, при дальнейшем заполнении интервала E3:E8 не будет изменяться.

2. Выделите блок ячеек E3:E8. В ячейку F4 введите формулу $=C2/$C8 . Нажмите сочетание клавиш **Ctrl + Enter**. Во всех ячейках выделенного диапазона появятся нужные формулы.

3. В ячейку F4 сразу введите правильную формулу $=C2/$C8 .

• Выделите ячейки D2:E8. Установите в них процентный формат, нажав на кнопку Процентный формат панели инструментов Форматирование. Увеличьте точность полученного значения до двух знаков после запятой. Для этого достаточно несколько раз щелкнуть по кнопкам Увеличить разрядность или Уменьшить разрядность панели инструментов Форматирование.



Величина премиального фонда каждого магазина определяется в размере 20% от положительной разницы между фактическим и плановым товарооборотом по каждой группе товаров, только в случае, если отношение фактического и планового товарооборота магазина по группе товаров превышает 100%. Премию можно рассчитать следующим способом:

• Установите курсор в ячейке F2.

• Щелкните по кнопке Вставка функции панели инструментов Стандартная.

• В категории Логические выберите функцию Если (рис.3.12.).

• В окне Логическое выражение введите условия выполнения плана свыше 100%. Для этого щелкните в ячейке D2 (% выполнения плана), перейдите на английскую раскладку клавиатуры, комбинацией клавиш **Shift + Ю** введите знак "больше", затем напишите 100%. Если, вместо % Вы напишете просто число 100, то оно будет умножено на 100. Поэтому, вместо 100% можно написать 1. Признаком правильного написания формулы является появление справа от окна текста **=ИСТИНА**.

• В окне Значение_если_истина напишите формулу, по которой 20% от перевыполнения плана товарооборота пойдет на премирование сотрудников $=0,2*(C2-B2)$. Признаком правильного написания формулы является появление справа от окна текста **= 28,2**.

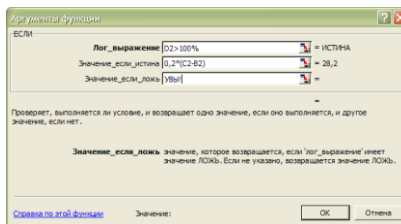


Рис. 3.12. Диалоговое окно функции Если

• В окне Значение_если_ложь, напечатайте какое-либо сочувственное выражение, например «Увы!».

Если все выполнено верно, то в диалоговом окне в двух местах появится значение 28,2

• Нажмите кнопку **ОК**. В ячейке появится найденное значение.

• Щелкните правой кнопкой мыши в ячейку F2. В появившемся контекстном меню выберите команду Копировать. Вокруг ячейки появится движущееся обрамление. Содержимое ячейки будет скопировано в буфер обмена.

• Выделите ячейки F3:F7, щелкните правой кнопкой мыши в любом месте выбранного диапазона. В появившемся контекстном меню выберите команду Вставить. Во всех ячейках диапазона появились значения.

В правом нижнем углу диапазона появится кнопка Параметры вставки. Ее действие аналогично действию с кнопкой **Параметры автозаполнения**.

• В ячейке F8 подсчитайте суммарное значение начисленной премии по всей фирме, как $=СУММ(F2:F7)$.

• Подсчитайте максимальное значение для каждого столбца. Вы уже выполнили такой подсчет для ячеек A2:A7. Для этого выполните копирование содержимого ячейки B9 в интервал ячеек C9:F9:

В ячейках появятся следующие значения (рис. 3.13). Видно, что формат чисел не тот, что Вам

8	Итого	2267	2533	111,73%	100,00%	63,6
9	Максимум	465	541	1,439252	0,213581	28,2

Рис. 3.13. Фрагмент расчетной таблицы без форматирования ячеек

нужен. Можно вновь выполнить операции форматирования, как это Вы уже делали, но воспользуемся кнопкой Формат по образцу панели инструментов Стандартная:

• Выделите интервал ячеек D8:F8. Щелкните левой кнопкой мыши на кнопке Формат по образцу панели инструментов Стандартная



- Выделите интервал ячеек D9:F9. Формат ячеек будет преобразован по выделенному образцу (рис.3.14.)

8	Итого	2267	2533	111,73%	100,00%	63,6
9	Максимум	465	541	143,93%	21,36%	28,2

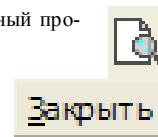
Рис. 3.14. Фрагмент расчетной таблицы после форматирования ячеек

☞ **Примечание.** Если Вы выполните двойной щелчок на кнопке Формат по образцу, то Вы имеете возможность многократного использования выбранного формата. Для отмены форматирования вновь щелкните на кнопке Формат по образцу. Вычисления закончены. Приступайте к оформлению таблицы.

ОФОРМЛЕНИЕ ТАБЛИЦЫ

Ячейки таблицы не имеют обрамления. Для проверки этого, щелкните левой кнопкой мыши на кнопке Предварительный просмотр панели инструментов Стандартная.

Вернуться на Рабочий лист можно, щелкнув на кнопке Закрыть, или нажав на клавишу **ESC**.



☞ **Выполнение задания.**

Для выполнения обрамления ячеек рабочего листа необходимо выполнить следующее:

- Выделите левой кнопкой мыши всю таблицу целиком, начиная с ячейки A1 до ячейки F9.
- Щелкните на стрелочке кнопки Границы панели инструментов Форматирование (рис. 3.15).

- Выберите обрамление каждой ячейки (вторая слева в нижней строке). Каждая ячейка выбранного блока будет обрамлена тонкой линией.
- Не снимая выделения, вновь откройте варианты обрамления, и выберите в них полужирное обрамление (правая в нижнем ряду). Вокруг таблицы будет проведена утолщенная линия.

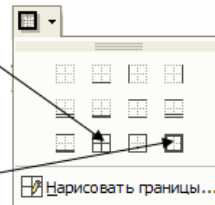


Рис. 3.15. Различные варианты обрамления ячеек

- Снимите выделение. Все ячейки таблицы будут обрамлены.

Выполните заливку шапки таблицы.

- Выделите ячейки A1:F1.
- Щелкните на стрелочке кнопки Цвет заливки панели инструментов Форматирование.

- Выберите цвет заливки (рис. 3.16). Каждая ячейка выбранного интервала залита выбранным цветом.

☞ **Примечание.** Не увлекайтесь темными цветами и не делайте слишком разноцветной таблицу. Далее, нам неоднократно понадобится выделение частей таблицы, а это будет затруднено, при ее излишней "попугаистости".

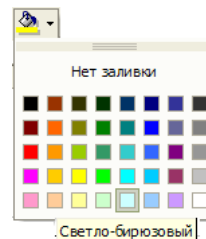


Рис. 3.16. Различные варианты заливки ячеек

Для придания более наглядного вида и подведения итогов, созданную таблицу целесообразно отсортировать.

3.10. СОРТИРОВКА ТАБЛИЦЫ

Строки таблицы можно отсортировать в алфавитном, числовом или хронологическом порядке не более чем по трем столбцам.

3.10.1. ПОРЯДОК СОРТИРОВКИ

Для каждого столбца электронной таблицы можно задать **возрастающий** или **убывающий** порядок сортировки.

1. **Возрастающий порядок сортировки:**

- Числа будут отсортированы от наименьшего отрицательного до наибольшего положительного числа.
- Значение даты и времени будут отсортированы от наиболее раннего, до наиболее позднего значения.
- Текст будет отсортирован по алфавиту.
- Логические значения будут отсортированы сначала по значению Ложь, а затем - по Истина.
- Ошибки будут отсортированы в порядке их появления.
- Пустые ячейки будут отображены в конце отсортированного списка.

2. **Убывающий порядок сортировки:**

- Строки электронной таблицы сортируются в обратном описанному выше порядку. Пустые ячейки по-прежнему будут размещены в конце списка.

Порядок сортировки определяется самим пользователем.

☞ **Примечание.** Перед началом сортировки **выделите диапазон сортируемых строк**, прихватив в него строку заголовка и **не включив строки, которые не нужно сортировать**, например, итоговые или заголовочные строки.

СОРТИРОВКА СТРОК ЭЛЕКТРОННОЙ ТАБЛИЦЫ

☞ **Выполнение задания.**

- Выделите диапазон сортируемых строк, в нашем случае это **блок** ячеек (п.3.5) A1:F7, а затем выберите команду Данные / Сортировка (рис. 3.17).

☞ **Примечание 1.** В случае простого щелчка автоматически будет выделена вся таблица в интервале A1:F9, а это неверно, так как в диапазон попадут строки Итого и Максимум.

☞ **Примечание 2.** Если справа от выделяемой таблицы отсутствуют вычисляемые ячейки, то для ускорения процесса выделения можно просто **выделить строки** (п.3.5) с первой по седьмую.

- В появившемся диалоговом окне Сортировка диапазона задайте параметры сортировки, изображенные на (рис. 3.18.) .

☞ **Примечание 1.** Если выполненное программой выделение диапазона не совсем корректно, установите переключатель окна диалога в нужное положение:

Идентифицировать поля по подписям (первая строка диапазона) или Идентифицировать поля по обозначениям столбцов листа.

- Результаты сортировки можно отменить, нажав на кнопку **Отмена**.

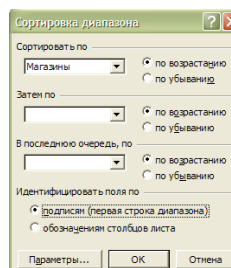
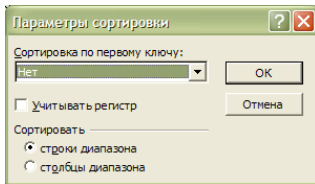


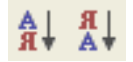
Рис. 3.17. Диалоговое окно Сортировка диапазона



⚡ **Примечание 2.** Для Вашей задачи это делать не надо, но для задания сортировки столбцов диапазона можете нажать в окне диалога Сортировка диапазона кнопку **Параметры** и в окне диалога Параметры сортировки в группе Сортировать установите переключатель в положение Столбцы диапазона (рис. 3.18).

Рис. 3.18. Диалоговое окно Параметры сортировки

⚡ **Примечание 3.** Если список нужно отсортировать только по одному полю, то порядок сортировки можно задать, нажав соответствующие кнопки на панели инструментов Стандартная. При этом столбец, по которому должна производиться сортировка, должен быть **самым левым в таблице** (как в Вашей задаче) и сортируются **все строки**. Данный вариант таблицы будет основным для всех последующих операций.



3.11. ПОДВЕДЕНИЕ ПРОМЕЖУТОЧНЫХ ИТОГОВ

Вам необходимо подвести промежуточные и общий итоги по столбцам Товароборот план, Товароборот факт, Премия, Доля для каждого магазина.

☞ Выполнение задания.

Для того чтобы не испортить имеющуюся таблицу, выполните ее копирование в том же Рабочем листе:

- Лево́й кнопкой мыши **выделите ячейки** A1:F9 или **строки** с первой по девятую.
- Щелкните правой кнопкой мыши в выделенном диапазоне ячеек. Выберите команду Копировать.
- Щелкните правой кнопкой мыши в ячейке A11. Выберите команду Вставить.
- В ячейках A11:F19 появится дубль основной таблицы (рис.3.19). Далее, работаем только с дублем.

	A	B	C	D	E	F
	Магазины	Товарооборот план, тыс.руб	Товарооборот факт, тыс.руб	% выполнения	Доля	Премия, тыс.руб
11		321	462	143.93%	0.18	28.2
12	1	465	541	116.34%	0.21	15.2
13	2	451	400	88.69%	0.16	УВЫШ
14	2	254	300	118.11%	0.12	9.2
15	3	325	380	116.92%	0.15	11
16	3	451	450	99.78%	0.18	УВЫШ
17	Итого	2267	2533	111.73%	1.00	63.6
18	Максимум	465	541	143.93%	0.21	28.2

Рис. 3.19. Дубль основной таблицы со всеми вычислениями и отсортированными строками

Перед подведением промежуточных итогов обязательно выполнение сортировки и желательное удаление лишних строк. В нашем случае сортировка по возрастанию номеров магазинов уже выполнена. Осталось удалить строки 18 – Итого и 19 – Максимум.

☞ Выполнение задания.

- Лево́й кнопкой мыши выделите номера строк 18 и 19 в заголовке строк. Строки будут закрашены темным цветом.
- Щелкните правой кнопкой мыши в выделенном диапазоне и выберите команду Удалить.

Строки 18 – Итого и 19 – Максимум будут удалены.

- Щелкните мышью в получившейся таблице или выделите ячейки A11:F17.
- Выберите команду Данные/Итоги.
- В диалоговом окне Промежуточные итоги (рис. 3.20) из списка При каждом изменении в выберите поле Магазины

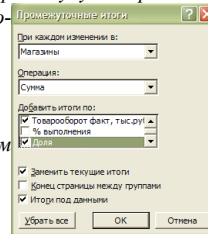


Рис. 3.20. Диалоговое окно Промежуточные итоги

- В списке Операция (рис. 3.21) укажите функцию, которая должна использоваться при определении промежуточных итогов. Если поле, по которому будут определены промежуточные итоги, содержит числовые значения, программа автоматически выберет из списка функцию Сумма. В противном случае это будет Количество значений.

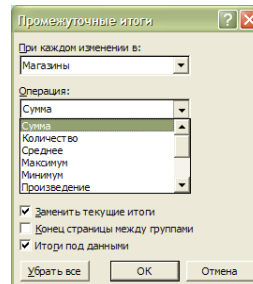


Рис. 3.21. Диалоговое окно Промежуточные итоги с раскрытым меню Операция

- В поле **Добавить итоги по** (рис. 3.20) укажите поля, по которым должны быть определены промежуточные итоги Товароборот план и Товароборот факт, а также Премия и Доля. При первом открытии диалога по умолчанию выбрано последнее поле Премия. При необходимости, перемещайтесь внутри окна с помощью стрелок перемещения.

По умолчанию промежуточные итоги будут отображены под отдельными группами записей, а общий итог в конце списка, но можете это изменить. Установите флажок **Заменять текущие итоги**, для последующих автоматических пересчетов результатов при изменении детальных данных.

- Если отдельные группы записей должны быть расположены и выведены на печать на различных страницах, установите флажок **Конец страницы между группами**. Для данного примера флажок не устанавливайте.
- Закройте окно диалога нажатием кнопки **ОК**.

Результаты подведения итогов приведены на рис. 3.22 и 3.23. Щелкните на кнопке "минус" в строке 1 Всего. Детальные данные будут скрыты. "Минус" превратился в "плюс". Промежуточные результаты будут удалены. Останется только итог по первому магазину.

- Щелкните по цифре один в левом верхнем углу списка. Будут сокрыты все детали и подведены итоги по всем организациям в целом (рис. 3.22).

	A	B	C	D	E	F
	Магазины	Товарооборот план, тыс.руб	Товарооборот факт, тыс.руб	% выполнения	Доля	Премия, тыс.руб
11						
+	21	Общий итог	2267	2533		1,00 63,6

Рис. 3.22. Результаты подведения итогов при сокрытии всех деталей

- Щелкните по цифре два в левом верхнем углу списка (рис. 3.23). Будут подведены итоги по каждому магазину и по всем магазинам в целом. Этот вариант таблицы Вам понадобится для разработки диаграммы.

	A	B	C	D	E	F
	Магазины	Товарооборот план, тыс.руб	Товарооборот факт, тыс.руб	% выполнения	Доля	Премия, тыс.руб
11						
+	14	1 Итого	786	1003		0.40 43.4
+	17	2 Итого	705	700		0.28 9.2
+	20	3 Итого	776	830		0.33 11
+	21	Общий итог	2267	2533		1,00 63,6

Рис. 3.23. Результаты подведения итогов при подведения второго уровня группировки

☞ **Примечание (посмотрите, но не выполняйте):**

Для удаления всех итогов в диалоговом окне Промежуточные итоги нажмите кнопку **Убрать все**. После этого, Вы можете подводить новые итоги.

3.12. СОЗДАНИЕ И РЕДАКТИРОВАНИЕ ДИАГРАММ

Одним из наиболее мощных средств анализа информации, содержащейся в таблицах данных, является **диаграмма**. Использование диаграмм позволяет выявить закономерности, мимо которых Вы могли бы пройти, не прибегая к их помощи. Это становится особенно ощутимым, когда производится анализ сложных процессов, таких, как деятельность любого предприятия. "Сухие" цифры отчета, безусловно, информативны и без них никакой анализ в принципе невозможен. Но только диаграмма позволяет проследить динамику процесса на его различных стадиях.

Для представления одной и той же информации можно использовать различные типы диаграмм, дополняющие друг друга и позволяющие вскрыть различные закономерности процесса.

☞ **Примечание.** Несмотря на все прелести использования диаграмм для анализа данных, к выбору их типа надо относиться продуманно. Иначе, как и всякое средство статистического анализа данных, оно может привести Вас к ложным выводам.

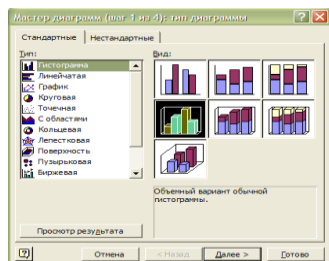
Для примера рассмотрим создание **объемной гистограммы** и **объемной круговой диаграммы**.

СОЗДАНИЕ ДИАГРАММЫ (ОБЪЕМНОЙ ГИСТОГРАММЫ)

☞ **Выполнение задания.**

Создайте диаграмму, сравнивающую плановые и фактические товарообороты по всем магазинам, ее форматирование до вида, указанного в задании:

- Выделите ячейки A11:C20 (итоги по столбцам Магазины, Товароборот план и Товароборот факт, вместе с заголовками)
- Нажмите кнопку Мастера диаграмм панели Стандартная или выполните команду Вставка / Диаграмма



- Выберите Объемный вариант обычной гистограммы (2 строка - 1 колонка - рис. 3.24). Нажмите кнопку **Далее**, и в следующем окне, вновь нажмите кнопку **Далее**.

Рис. 3.24. Первое диалоговое окно Мастера диаграмм - выбор типа диаграммы

- Перейдите на вкладку **Заголовки** (рис.3.25). Введите заголовки. В окне **Название диаграммы** напечатайте **АНАЛИЗ ВЫПОЛНЕНИЯ ПЛАНА Т/О**. Заголовки для осей X и Y не заполняйте / **Далее**

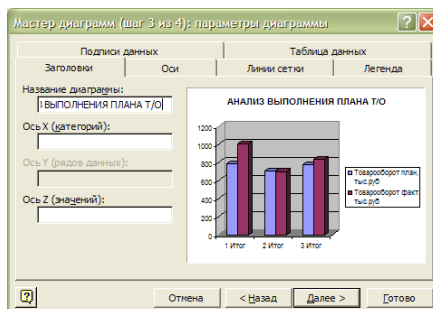


Рис. 3.25. Третье диалоговое окно Мастера диаграмм - задание заголовков в диаграмме

- Вкладка **Легенда** (рис. 3.26). Выберите расположение **Справа**. / кнопка **Далее**
- Остальные вкладки не трогайте. Нажмите кнопку **Готово**.
- Поместите диаграмму на имеющемся листе. / Нажмите кнопку **Готово**.

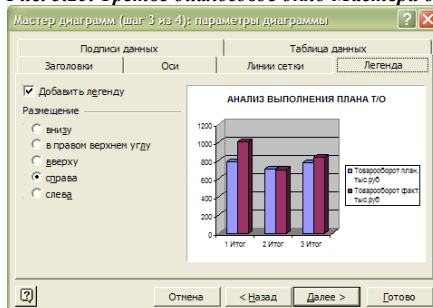


Рис. 3.26. Третье диалоговое окно Мастера диаграмм - вкладка Легенда

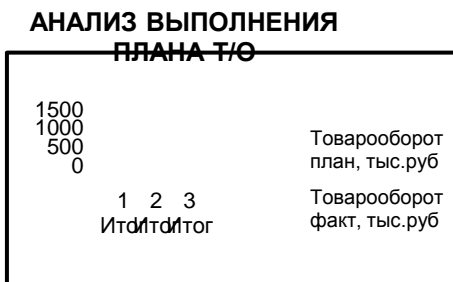
- Переместите указатель мыши на диаграмму, пока не появится сообщение **Область диаграммы**. Нажмите левую кнопку мыши и переместите диаграмму ниже таблицы.

- С помощью маркеров расположите диаграмму в области, не превышающей A22:F36.

Диаграмма будет иметь вид, приведенный на рис.

3.27.

Рис. 3.27. Внешний вид полученной диаграммы



РЕДАКТИРОВАНИЕ ДИАГРАММЫ (выполнять по указанию преподавателя)

- **Уберите фон за столбцами товарооборотов.** Подведите курсор мыши в **Область построения диаграммы** (прямоугольник серого цвета), пока не появится текст **Стенки**. Щелкните правой кнопкой и выберите команду **Очистить**.
- **Установите новый формат обозначения магазинов.** Для этого щелкните в области **Оси категорий** правой кнопкой мыши. Выберите команду **Формат оси**. Щелкните по вкладке **Шрифт**. Установите шрифт 8 пт., обычный. ОК.
- **Установите новый формат области Ось значений** - шрифт 8 пт., полужирный.

- **Выполните заливку** Области диаграммы. Щелкните правой кнопкой в Области диаграммы / откройте вкладку Вид / Формат области диаграммы (рис. 3.28). Выберите любой из цветов (предпочтительно не очень темный) / ОК.

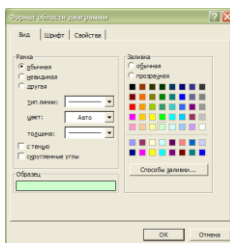


Рис. 3.28. Диалоговое окно Формат области построения - выбор заливки Области диаграммы

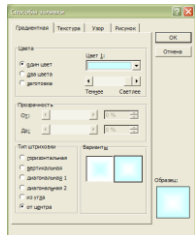
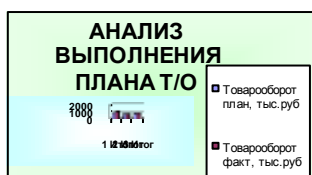


Рис. 3.29. Диалоговое окно Заливка - выбор заливки Области построения



СОЗДАНИЕ ОБЪЕМНОЙ КРУГОВОЙ ДИАГРАММЫ (выполнять по указанию преподавателя)

Теперь создайте объемную круговую диаграмму удельных весов каждого магазина в суммарном фактическом товарообороте всех магазинов.

Назначение круговых диаграмм - показ удельного веса составляющих в общей совокупности.

Диаграмма (после ряда корректировок) может иметь следующий вид (3.31):

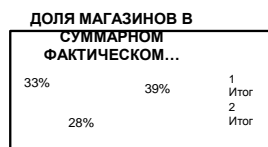


Рис. 3.31. Внешний вид разрезанной объемной круговой диаграммы

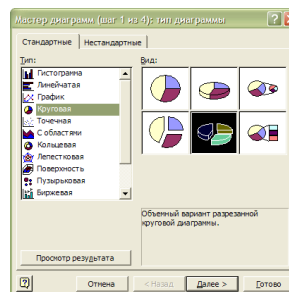
Выполнение задания.

- Выделите ячейки A11:A20 (столбец Магазины), а затем при нажатой клавише **Ctrl** выделите ячейки C11:C20 Товарооборот факт.
- Нажмите кнопку Мастера диаграмм или выполните команду Вставка / Диаграмма



- Выберите Объемный вариант разрезанной круговой диаграммы (2 строка - 2 колонка) (рис. 3.32) / кнопка Далее / кнопка Далее

Рис. 3.32. Первый шаг мастера диаграмм - выбор типа диаграммы



- Вкладка Заголовки (рис. 3.33). Введите заголовки. В окне Название диаграммы, напечатайте текст ДОЛЯ МАГАЗИНОВ В СУММАРНОМ ФАКТИЧЕСКОМ ТОВАРООБОРОТЕ.
- Вкладка Подписи данных. В группе переключателей Подписи значений включите Доля / кнопка Далее

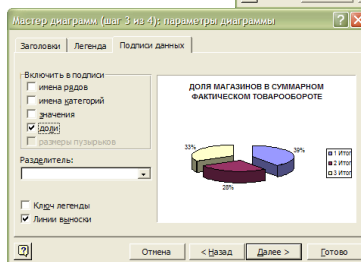


Рис. 3.33. Третий шаг мастера диаграмм - установка вида подписей на диаграмме

- Поместите диаграмму на имеющемся листе. Нажмите кнопку Готово.
- Переместите указатель мыши на диаграмму, пока не появится сообщение Область диаграммы. Нажмите левую кнопку мыши и переместите диаграмму ниже созданных ранее диаграмм в интервале ячеек A38:F50.

СВОДНЫЕ ТАБЛИЦЫ (выполнять по указанию преподавателя)

Сводная таблица представляет собой интерактивную таблицу, предназначенную для анализа больших массивов данных.

Панель инструментов Сводная таблица (рис. 3.34):

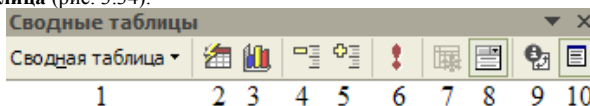


Рис. 3.34. Панель инструментов Сводная таблица

- | | | | |
|---|------------------------|----|-------------------------------------|
| 1 | Вызов выпадающего меню | 7 | Обновить данные |
| 2 | Формат отчета | 8 | Учитывать скрытые элементы в итогах |
| 3 | Мастер диаграмм | 9 | Всегда отображать элементы |
| 4 | Скрыть детали | 10 | Параметры поля |
| 5 | Отобразить детали | 11 | Скрыть список полей |

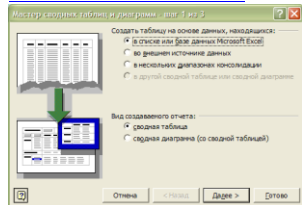
СОЗДАНИЕ СВОДНОЙ ТАБЛИЦЫ

☞ **Выполнение задания.**

Создайте таблицу, в которой в заголовках столбцов будут располагаться магазины, а в ячейках столбцов - сумма проданного товара. Для создания сводной таблицы возвращаемся к исходной таблице, полученной в п. 3.11.

Чтобы создать сводную таблицу:

Выделите диапазон ячеек A1:C7



- Укажите источник данных.

По умолчанию переключатель установлен в положении в списке или базе данных Microsoft Excel. Оставьте его неизменным.

- Нажмите кнопку **Далее**.

На втором шаге (рис. 3.36) Вы должны проверить ссылку на диапазон с исходными данными, предлагаемый Мастером. При необходимости измените диапазон с помощью мыши.

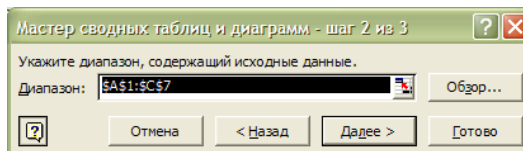


Рис. 3.36. Второй шаг мастера Сводных таблиц - указание диапазона, содержащего исходные данные

После нажатия кнопки **Обзор** Вы получаете доступ к закрытым рабочим книгам.

☞ **Примечание.** Если на этом этапе появилось сообщение об ошибке Недопустимое имя поля сводной таблицы, проверьте, не содержит ли первая строка выделенного диапазона пустые значения.

На третьем шаге вам надо щелкнуть по кнопке Макет... (рис. 3.37).

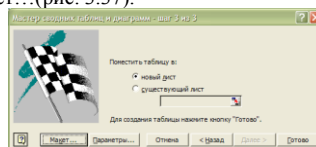


Рис. 3.36. Третий шаг мастера Сводных таблиц - указание создания макета сводной таблицы

- Определите макет сводной таблицы (рис. 3.38). Перетащите левой кнопкой мыши поле Магазины в область заголовков столбцов, поля Товарооборот план и Товарооборот факт в область данных. Поле страницы используется для фильтрации данных по задаваемому у него имени поля (пока не трогайте).

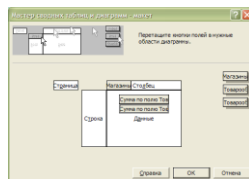


Рис. 3.38. Определение макета сводной таблицы

- Нажмите кнопку **OK**.
- Вы вновь попадаете на [третий шаг Мастера сводных таблиц](#) (рис. 3.36). Сводную таблицу можно разместить на любом рабочем листе любой рабочей книги.
- Поместите таблицу в новом рабочем листе, нажав на кнопку **ГОТОВО**. Назовите его **СВОДНАЯ ТАБЛИЦА**.
- Для этого дважды щелкните левой кнопкой мыши на [ярлыке листа](#) (п.3.4) и напишите текст "Сводная таблица".

Фрагмент таблицы, приведен ниже (рис. 3.39).

	A	B	C	D	E
3		Магазины			
4	Данные	1	2	3	Общий итог
5	Сумма по полю Товарооборот план, тыс.руб	786	705	776	2267
6	Сумма по полю Товарооборот факт, тыс.руб	1003	700	830	2533

Рис. 3.39. Изображение фрагмента первой таблицы, сформированной с помощью операции **Закрепление областей** ИЗМЕНЕНИЕ ВИДА ДАННЫХ В СВОДНОЙ ТАБЛИЦЕ

☞ **Выполнение задания.**

Чтобы переместить поля данных:

Измените созданную сводную таблицу таким образом, чтобы по строкам в ней стали располагаться данные о магазинах.

- Щелкните указателем мыши в любом месте области сводной таблицы. Появится панель инструментов [Сводные таблицы](#) и окно [Список полей сводной таблицы](#) (рис. 3.40).
- Щелкните на поле Магазины, а затем на кнопке **Поместить в**. Поле будет перемещено в область строк. В списке можно выбрать и другое место расположения перемещаемого поля.

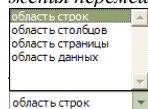


Рис. 3.40. Диалоговое окно перемещения полей сводной таблицы

	A	B	C
3	Данные	Магазины	Итог
4	Сумма по полю Товарооборот план, тыс.руб	1	786
5		2	705
6		3	776
7	Сумма по полю Товарооборот факт, тыс.руб	1	1003
8		2	700
9		3	830
10	Итог Сумма по полю Товарооборот план, тыс.руб		2267
11	Итог Сумма по полю Товарооборот факт, тыс.руб		2533

Рис. 3.41. Изображение фрагмента второй таблицы, сформированной с помощью операции **Закрепление областей**

☞ **Примечание.** Перемещение полей можно проводить непосредственно в самой сводной таблице. Так, если вы захватите поле Магазины и переместите его в район "нулевой ячейки" (п. 3.5), то внешний вид Сводной таблицы примет, следующий вид (рис. 3.42).

- Выберите команду **Данные / Сводная таблица**. На экране появится первое диалоговое окно Мастера сводных таблиц (рис.3.35).

Рис. 3.35. Первый шаг мастера Сводных таблиц - указание места нахождения источника данных

- При перемещении поля изменится не только макет сводной таблицы, а также вид данных и результаты (рис. 3.41).

	A	B
1	Магазины	(Все) ▾
2		
3	Данные ▾	Итого
4	Сумма по полю Товарооборот план, тыс.руб	2267
5	Сумма по полю Товарооборот факт, тыс.руб	2533

Рис. 3.42. Изображение фрагмента третьей таблицы, полученной перемещением поля Магазины в область страницы непосредственно мышью

ПРЕДВАРИТЕЛЬНЫЙ ПРОСМОТР И ПЕЧАТЬ СОДЕРЖИМОГО ОКНА РАБОЧЕЙ КНИГИ

Просмотр и печать в Excel XP и Word XP очень похожи.

Прежде, чем напечатать содержимое, следует [посмотреть](#), как он будет выглядеть в напечатанном виде. По умолчанию Excel XP выведет на печать содержимое всего рабочего листа. Вы можете предварительно [выделить область печати](#) и вывести на печать только указанную область.

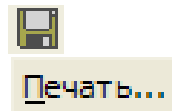
☛ Выполнение задания.

- Перейдите на первый лист рабочей книги. Нажмите на кнопку Предварительный просмотр панели инструментов Стандартная. Лист отобразится в том виде, в котором он будет напечатан (рис. 3.43). Возможностей изменения внешнего вида просматриваемого экрана значительно меньше. На первом этапе работы вам достаточно просто изменять масштаб изображения. Для перехода в обычный режим достаточно щелкнуть по кнопке Закрыть.

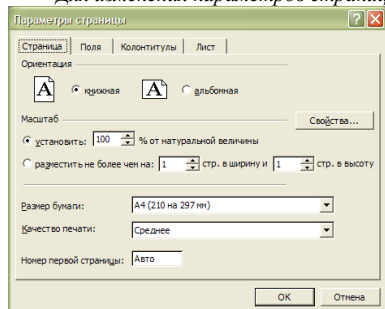


Рис. 3.43. Экран предварительного просмотра окна рабочей книги Excel XP

- Для того чтобы напечатать документ, достаточно щелкнуть по кнопке Печать на панели инструментов Стандартная или Предварительный просмотр. Документ будет напечатан на заданном по умолчанию принтере с использованием заданных по умолчанию параметров.



- Для изменения параметров страницы выберите команду Файл / Параметры страницы (или кнопку Страница из режима просмотра).



В появившемся диалоговом окне Параметры страницы (рис. 3.44) можно указать формат бумаги, размеры полей и ориентацию страницы

Рис. 3.44. Определение формата бумаги

В вашем распоряжении есть полезная возможность, отсутствующая в Word XP. Вы можете увеличить или уменьшить масштаб изображения таким образом, чтобы он вошел целиком на один лист.

Остальные возможности диалогового окна Параметры страницы вы можете изучить самостоятельно.

- Для установки новых параметров печати перейдите командой Файл / Печать в окно Печать (рис. 3.45).

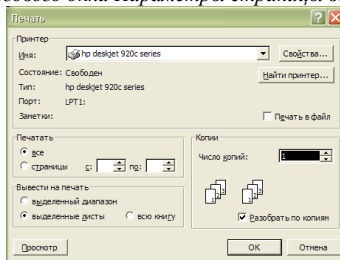


Рис. 3.45. Диалоговое окно Печать Word XP

- Если нужно указать другой принтер, щелкните на стрелке поля Принтер имя, выберите принтер из списка.
- Если ваши данные размещаются на нескольких страницах рабочего листа, то можете указать их.
- В нашем случае этого делать не следует, но вы можете установить необходимые параметры печати, используя кнопки диалогового окна.
- Если к вашему компьютеру подключен принтер, то можете напечатать документ, щелкнув по кнопке **ОК**.

КОПИРОВАНИЕ ТАБЛИЦ И ДИАГРАММ ИЗ EXCEL XP В WORD XP (создание динамической связи)

При выполнении операции копирования (вырезания) объекта и последующей его вставки в другое приложение MS Office объект вставляется в виде "моментальной фотографии".

☛ Выполнение задания.

Для создания *связи* между таблицами и диаграммами Excel XP и текстом Word XP можно выполнить обычные операции копирования и вставки, но лучше использовать механизм *Специальной вставки*.

☞ **Примечание.** Операции связывания объектов Excel XP и Word XP выполняются только отдельно для каждого объекта (таблицы или диаграммы).

- **Выделите таблицу** (блок ячеек A1:F9).
- Выполните любым способом ее **копирование в буфер обмена**, например, щелкните правой кнопкой мыши в выделенном диапазоне ячеек, и выберите команду Копировать.
- Перейдите в документ Word XP.
- Установите курсор в место вставки объекта.
- Выделите команду Правка / Специальная вставка (рис. 3.46).
- Установите флажок Связать.

- Выберите вариант вставки.

☞ **Примечание.** Можете оставить связь такой, как ее предлагает Excel XP, но тогда вам, в дальнейшем, нужно будет редактировать связанные объекты средствами Word XP. Для точной копии таблицы Excel XP лучше выбрать опцию Лист Microsoft Excel (объект). В этом случае, перемещение таблицы и изменение ее размеров будет проще.

- Нажмите кнопку **ОК**.

Такие операции повторите для интервала ячеек A11:F21, при нажатой кнопке под номером "2", а так же для одной из диаграмм.

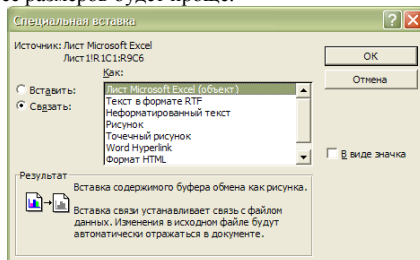


Рис. 3.46. Диалоговое окно Специальная вставка

☞ **Примечание.** Для выделения диаграммы достаточно просто щелкнуть в Области диаграммы левой или правой кнопкой мыши.

☞ **Выполнение задания.**

Выполните форматирование связанных таблиц и диаграмм на листе Word с максимальным подобием заданию.

Для этого, возможно, понадобится задание правильного обтекания для любого объекта. Рассмотрим данную операцию на примере диаграммы.

Если, при попытке установки диаграммы справа, от ранее установленной таблицы, Ваша диаграмма перемещается выше или ниже таблицы, то:

- Щелкните правой кнопкой мыши на границе Области диаграммы (курсор примет вид белой стрелки с четырьмя маленькими черными стрелочками).
- В появившемся контекстном меню выберите команду **Формат объекта**.
- В диалоговом окне **Формат объекта** (рис. 2.13. п 2.5.2) выберите вкладку **Обтекание**.
- Установите **Обтекание вокруг рамки**.
- Щелкните на кнопку **ОК**.
- **Левой кнопкой мыши переместите диаграмму так, как это сделано в задании.**

Если после нескольких безуспешных попыток сгруппировать объекты, можете:

- Ниже основного текста **создать таблицу** размером две строки на два столбца.
- В каждую ячейку **вставить** или **связать** по объекту (вместо команды **Копировать** задать команду **Вырезать**). Команды **Копировать** (Вырезать) объект в буфер обмена и **Вставить содержимое буфера** для любого приложения Windows XP выполняются аналогично.

☞ **Примечание.** Именно таким образом вставлено большинство рисунков в текст данного пособия

- **Сохраните** созданный документ Word XP и лист Excel XP. Место сохранения - Ваша папка на диске (или Ваша дискета).

ВОПРОСЫ ДЛЯ САМОПОДГОТОВКИ

1. Как запустить Excel XP? [📖](#)
2. Какие новые элементы листа введены в Excel XP по сравнению с предыдущими версиями? [📖](#)
3. Существуют ли принципиальные отличия при сохранении Excel XP и Word XP? [📖](#)
4. Что такое рабочий лист Excel XP? [📖](#)
5. Каким образом нумеруются ячейки в Excel XP? [📖](#)
6. Укажите способ быстрого перемещения между листами рабочей книги в Excel XP? [📖](#)
7. Каким способом осуществляется выделение строк, столбцов, блока ячеек, целого листа в Excel XP? [📖](#)
8. Как выглядит и где находится маркер заполнения в Excel XP? [📖](#)
9. Как выделить группу ячеек с помощью мыши и клавиатуры в Excel XP? [📖](#)
10. Как выделить две группы ячеек, разнесенных в пространстве таблицы в Excel XP? [📖](#)
11. Каким образом можно завершить ввод данных в ячейку рабочего листа в Excel XP? [📖](#)

Тема 2.4. Архиваторы. Антивирусы (лекция – дискуссия 1 час)

Архиваторы

Архиватор — программа, осуществляющая **сжатие** и/или упаковку одного и более **файлов** в **архив** или серию архивов для удобства переноса или хранения, а также распаковку архивов.

Простейшие архиваторы просто последовательно объединяют (упаковывают) содержимое файлов в архив. Архив должен также содержать информацию об именах и длине оригинальных файлов для их восстановления, поэтому большинство архиваторов также сохраняют **метаданные** файлов, предоставляемые **операционной системой**, такие, как время создания и права доступа. Многие архиваторы используют **сжатие без потерь** для уменьшения размера архива.

Характеристики архиваторов:

- По степени сжатия.
- По скорости сжатия.

Эти характеристики — обратно зависимые величины. То есть, чем больше скорость сжатия, тем меньше степень сжатия, и наоборот.

Сжатие данных

Программа, создавая архив, обрабатывает как **текстовые файлы**, так и **бинарные файлы**. Первые всегда сжимаются в несколько раз (в зависимости от архиватора), тогда как сжатие бинарных файлов зависит от их характера. Одни бинарные файлы могут быть сжаты в десятки раз, сжатие же других может и вовсе не уменьшить занимаемый ими объём.

Нахождение для любого входного файла программы наименьшего возможного размера, печатающей этот файл, является **алгоритмически неразрешимой задачей**, поэтому «идеальный» архиватор невозможен.

Сжатие данных обычно происходит значительно медленнее, чем обратная операция.

Вирусы. Классификация вирусов

Что такое «компьютерный вирус»?

Компьютерный вирус — это специально написанная, как правило, небольшая по размерам программа, которая может записывать (внедрять) свои копии (возможно, измененные) в компьютерные программы, расположенные в исполнимых файлах, системных областях дисков, драйверах, документах и т.д., причем эти копии сохраняют возможность к «размножению».

Процесс внедрения вирусом своей копии в другую программу (системную область диска и т.д.) называется **заражением**, а программа или объект, содержащий вирус — **зараженным**.

Компьютерные вирусы являются программами, которые могут «размножаться» и скрытно внедрять свои копии в файлы, загрузочные секторы дисков и документы.

Обязательным свойством компьютерного вируса является способность к размножению (самокопированию) и незаметному для пользователя внедрению в файлы, загрузочные секторы дисков и документы. Название «вирус» по отношению к компьютерным программам пришло из биологии именно по признаку способности к саморазмножению.

Стадии развития вируса:

- **скрытый этап** — действие вируса не проявляется и остается незамеченным
- **лавинообразное размножение**, но его действия пока не активизированы

- **активные действия** – выполняются вредные действия, заложенные его автором.

Активизация вируса может быть связана с различными событиями:

- Наступлением определённой даты или дня недели
- Запуском программы
- Открытием документа и т.д.

История возникновения компьютерных вирусов

Основы теории самовоспроизводящихся механизмов заложил американец венгерского происхождения Джон фон Нейман, который в 1951 году предложил метод создания таких механизмов.

Первой публикацией была посвящённой созданию самовоспроизводящихся систем, является статья Л. С. Пенроуз в соавторстве со своим мужем, нобелевским лауреатом по физике Р. Пенроузом, о самовоспроизводящихся механических структурах, опубликованная в 1957 году американским журналом Nature.

В этой статье, наряду с примерами чисто механических конструкций, была приведена некая двумерная модель подобных структур, способных к активации, захвату и освобождению.

С 1961 года известны рабочие примеры таких программ.

В 1961 году В. А. Высотский, Х. Д. Макилрой и Роберт Моррис из фирмы Bell Telephone Laboratories (США) изобрели необычную игру «Дарвин» (Darwin), в которой несколько ассемблерных программ, названных «организмами», загружались в память компьютера.

Организмы, созданные одним игроком, должны были уничтожать представителей другого вида и захватывать жизненное пространство. Победителем считался тот игрок, чьи организмы захватывали всю память или набирали наибольшее количество очков.

С появлением первых персональных компьютеров **Apple** в **1977 году** и развитием сетевой инфраструктуры начинается новая эпоха истории вирусов.

Появились первые программы-вандалы, которые под видом полезных программ выкладывались на BBS, однако после запуска уничтожали данные пользователей.

В это же время появляются троянские программы-вандалы, проявляющие свою деструктивную сущность лишь через некоторое время или при определённых условиях.

Первые вирусные эпидемии

Очередным этапом развития вирусов считается 1987 год.

К этому моменту получили широкое распространения сравнительно дешёвые компьютеры IBM PC, что привело к резкому увеличению масштаба заражения компьютерными вирусами.

Именно в 1987 вспыхнули сразу три крупные эпидемии компьютерных вирусов.

Первая эпидемия была вызвана вирусом **BRAIN** (также известен как Пакистанский вирус) - первый вирус созданный для IBM PC-совместимых ПК. Он был разработан братьями **Амджадом и Базитом Алви** в 1986 и был обнаружен летом 1987.

Программа должна была наказать местных пиратов, ворующих программное обеспечение у их фирмы. В программке значились имена, адрес и телефоны братьев. Однако неожиданно для всех The Brain вышел за границы Пакистана и заразил только в США более 18 тысяч компьютеров и сотни компьютеров по всему миру. Вирус Brain являлся также и первым стелс-вирусом — при попытке чтения заражённого сектора он «подставлял» его незаражённый оригинал.

Вторая эпидемия, берущая начало в Лехайском университете (США), разразилась в ноябре. В течение нескольких дней этот вирус уничтожил содержимое нескольких сот дискет из библиотеки вычислительного центра университета и личных дискет студентов. За время эпидемии вирусом было заражено около четырёх тысяч компьютеров.

Третья вирусная эпидемия разразилась перед самым Новым годом, 30 декабря. Её вызвал вирус, обнаруженный в Иерусалимском Университете (Израиль). Хотя существенного вреда этот вирус не принёс, он быстро распространился по всему миру.

В пятницу 13 мая 1988 сразу несколько фирм и университетов нескольких стран мира «познакомились» с вирусом Jerusalem — в этот день вирус уничтожал файлы при их запуске.

В ноябре 1988 году Роберт Моррис младший создает Червь Морриса, который инфицировал подключенные к Интернету компьютеры VAX, DEC и Sun под управлением ОС BSD. Червь Морриса стал первым сетевым червем, успешно распространившимся «in-the-wild».

Проблема вирусов начинает принимать **глобальный размах**.

1990 год выходит первый полиморфный вирус - **Chameleon**.

1992 год вирус Michelangelo порождает волну публикаций в западных СМИ, предсказывающих катастрофу 6-го марта. Ожидалось, что вирус повредит информацию на миллионах компьютеров, но его последствия оказались минимальными.

1992 год известен как год появления первых конструкторов вирусов, а также готовых полиморфных модулей и модулей шифрования. Начиная с этого момента, каждый программист мог легко добавить функции шифрования к своему вирусу.

В начальный период существования вредоносных программ были популярны вирусы-шутки, которые мешали работе пользователей. Деструктивные особи практически не встречались. Например, такие программы просили дополнительной памяти («пирожка» и т. п.), и экран блокировался, пока пользователь не вводил с клавиатуры нужное слово (иногда его нужно было угадать). Или, например, вирус, который выводил на экран сообщение вроде: «Нажмите одновременно **L + A + M + E + R + F1 + Alt**». Пользователь нажимал, после чего появлялось сообщение о том, что таблица разделов стерта с жесткого диска и загружена в оперативную память и если пользователь отпустит хотя бы одну клавишу, то со своей информацией он может проститься, а если просидит так ровно час, то все будет в порядке. Через час оказывалось, что это была шутка. Ничего себе шуточки...

Массовое распространение персональных компьютеров привело к появлению людей, которых интересовал не сам процесс создания вируса, а результат, наносимый вредоносной программой. Изменились и шутки: вирусы начали реально форматировать диски, стирать или кодировать важную информацию. Некоторые приложения использовали недостатки оборудования и портили его, например выстраивали лучи монитора в одну точку, прожигая его (надёжность устройств в то время оставляла желать лучшего), или ломали жесткие диски, гоня считывающую головку по определённому алгоритму.

На сегодня в мире вирусов наметилась явная коммерциализация: данные в большинстве случаев не уничтожают, а пытаются украсть. Например, популярное онлайн-игр привела к появлению вирусов, специализирующихся на краже паролей к учетным записям игроков, так как виртуальные ценности, заработанные в игре, можно продать за реальные деньги.

По "деструктивному воздействию" все вирусы можно разделить на:

Безвредные вирусы. Они не мешают работе компьютера, но могут уменьшать объем свободной оперативной памяти и памяти на дисках, действия таких вирусов проявляются в каких-либо графических или звуковых эффектах.

Опасные вирусы. К ней относятся вирусы, которые могут привести к определенным сбоям в работе операционной системы или некоторых программ.

Очень опасные вирусы. Эти вирусы могут уничтожить определенные или все данные, находящиеся на жестком диске, изменить системную информацию, вывести из строя операционную систему и т. д.

По "способу заражения" все вирусы можно разделить на:

Резидентные вирусы. Чаще всего эти вирусы являются одной из разновидностей файловых и загрузочных. Причем самой опасной разновидностью.

Резидентный вирус при заражении (инфицировании) компьютера оставляет в оперативной памяти свою резидентную часть, которая потом перехватывает обращение операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т. п.) и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера.

Нерезидентные вирусы не заражают память компьютера и являются активными ограниченное время.

По "среде обитания" все вирусы можно разделить на:

Файловые вирусы. До появления Интернета именно эти вирусы были самыми распространенными. На сегодняшний день известны зловредные программы, заражающие все типы выполняемых объектов любой операционной системы (для Windows опасности подвергаются исполняемые файлы (.exe, .com), командные файлы (.bat), драйвера (.sys), динамические библиотеки (.dll) и т. д.)

Заражение происходит следующим образом. Вирус записывает свой код в файл-жертву. Кроме того, зараженный файл специальным образом изменяется. В результате при обращении к нему операционной системы (запуск пользователем, вызов из другой программы и т. п.) управление передается в первую очередь коду вируса, который может выполнить любые действия, заданные ему создателем. После выполнения своих действий вирус передает управление программе, которая выполняется нормальным образом. Так что если на компьютере пользователя не установлено специальное ПО, он может долго не догадываться о заражении.

Загрузочные вирусы. Эти вирусы заражают загрузочные сектора жестких дисков.

Принцип их действия заключается в следующем. Вирус добавляет свой код к одной из специальных программ, которые начинают выполняться после включения компьютера до загрузки операционной системы. В принципе в задачу этого ПО как раз и входят подготовка и запуск ОС. Таким образом, вирус получает управление и может выполнить определенные действия, например записать себя в оперативную память. И только после этого будет загружаться операционная система. Вот только вирус уже будет находиться в памяти и сможет контролировать ее работу.

Макровирусы. Эти вирусы представляют собой программы, которые выполнены на языках, встроенных в различные программные системы. Чаще всего жертвами становятся файлы, созданные различными компонентами Microsoft Office (Word, Excel и т.д.). Встроенный в эти программные продукты Visual Basic прекрасно подходит для написания макровирусов.

Принцип их действия очень прост. Вирус записывает себя в DOT-файл, в котором содержатся все глобальные макросы, часть из которых он подменяет собой. После этого все файлы, сохраненные в этой программе, будут содержать макровирус. При этом он может выполнять множество различных деструктивных действий вплоть до удаления всех документов.

Сетевые вирусы. Главной особенностью этих вирусов является возможность работы с различными сетевыми протоколами. То есть они могут различными путями записывать свой код на удаленном компьютере. Наибольшее распространение в наше время получили интернет-черви. Эти вирусы чаще всего используют для своей работы электронную почту, "прицепляясь" к письму. При этом на новом компьютере они либо автоматически выполняются, либо различными способами подталкивают пользователя к своему запуску.

По "особенностям алгоритмов" все вирусы можно разделить на:

По особенностям алгоритма вирусы трудно классифицировать из-за большого разнообразия.

Простейшие вирусы - паразитические, они изменяют содержимое файлов и секторов диска и могут быть достаточно легко обнаружены и уничтожены. Можно отметить вирусы-репликаторы, называемые червями, которые распространяются по компьютерным сетям, вычисляют адреса сетевых компьютеров и записывают по этим адресам свои копии.

Стелс-вирусы (stealth – невидимка). Это вирусы, которые умеют скрывать свое присутствие в системе. Обнаружить их очень сложно. Стелс-вирусы используют различные способы обеспечения "невидимости". Наиболее распространен следующий вариант. Вирус состоит из двух частей. Одна из них резидентная и постоянно находится в памяти компьютера. Если операционная система обращается к зараженному файлу, то "резидент" перехватывает это обращение и удаляет из файла код вируса. Таким образом, приложение оказывается "чистым". Но после того как оно завершает свою работу, "резидент" снова заражает его.

Полиморфные вирусы. Особенностью этих вирусов является умение изменять собственный код. Сделано это для того, чтобы ввести в заблуждение антивирусные программы, часто использующие "маски" (отрывки кода, типичные для вирусов). Полиморфные вирусы бывают двух типов. Первые просто шифруют собственное "тело" с непостоянным ключом и случайным набором команд дешифратора. Вторая группа сложнее. Вирусы, относящиеся к ней, могут переписывать свой код, то есть, по сути, они сами являются программистами.

Троянский конь – это программа, содержащая в себе некоторую разрушающую функцию, которая активизируется при наступлении некоторого условия срабатывания. Обычно такие программы маскируются под какие-нибудь полезные утилиты.

«Троянские кони» представляют собой программы, реализующие помимо функций, описанных в документации, и некоторые другие функции, связанные с нарушением безопасности и деструктивными действиями. Отмечены случаи создания таких программ с целью облегчения распространения вирусов. Списки таких программ широко публикуются в зарубежной печати. Обычно они маскируются под игровые или развлекательные программы и наносят вред под красивые картинки или музыку.

Если вирусы и «тройные кони» наносят ущерб посредством лавинообразного саморазмножения или явного разрушения, то основная функция вирусов типа «червь», действующих в компьютерных сетях, – взлом атакуемой системы, т.е. преодоление защиты с целью нарушения безопасности и целостности.

Руткиты представляют собой более продвинутый вариант троянских коней. Некоторые антивирусные компании не разделяют руткиты и троянцы, относя их к одной категории зловредных программ. Однако троян прячется на компьютере, обычно маскируясь под известную программу (например, SpyMaster выдает себя за приложение MSN Messenger), а руткиты используют для маскировки более продвинутые методы, внедряясь глубоко в систему.

Изначально словом «руткит» обозначался набор инструментов, позволяющий злоумышленнику возвращаться во взломанную систему таким образом, чтобы системный администратор не мог его видеть, а система – регистрировать. Долгое время руткиты были привилегией Unix-систем, но, как известно, хорошие идеи просто так не пропадают, и в конце XX века стали массово появляться руткиты, предназначенные для Microsoft Windows.

Ботнет (*botnet* произошло от слов *robot* и *network*) — это компьютерная сеть, состоящая из некоторого количества хостов, с запущенными ботами — автономным программным обеспечением.

Чаще всего бот в составе ботнета является программой, скрытно устанавливаемой на устройство жертвы и позволяющей злоумышленнику выполнять некие действия с использованием ресурсов зараженного компьютера.

Обычно используются для нелегальной или неодолимой деятельности — рассылки спама, перебора паролей на удаленной системе, атак на отказ в обслуживании.

Червями называют вирусы, которые распространяются по глобальным сетям, поражая целые системы, а не отдельные программы. Это самый опасный вид вирусов, так как объектами нападения в этом случае становятся информационные системы государственного масштаба. С появлением глобальной сети Internet этот вид нарушения безопасности представляет наибольшую угрозу, т. к. ему в любой момент может подвергнуться любая из 40 миллионов компьютеров, подключенных к этой сети.

Бэкдор, backdoor (*back door* - чёрный ход) — программы, которые устанавливает взломщик на взломанном им компьютере после получения первоначального доступа с целью повторного получения доступа к системе.

Основное назначение Backdoor – скрытное управление компьютером. Как правило, Backdoor позволяет копировать файлы с пораженного компьютера и наоборот, передавать на пораженный компьютер файлы и программы. Кроме того, обычно Backdoor позволяет получить удаленный доступ к реестру, производить системные операции (перезагрузку ПК, создание новых сетевых ресурсов, модификацию паролей и т.п.).

Кейлоггер (*keylogger* — это регистратор нажатий клавиш) — программное обеспечение, основным назначением которого является скрытый мониторинг нажатий клавиш и ведение журнала этих нажатий или аппаратные средства. Аппаратные кейлоггеры встречаются значительно реже, чем программные, однако при защите важной информации о них ни в коем случае нельзя забывать.

Перехват нажатий клавиш может использоваться обычными программами и часто применяется для вызова функций программы из другого приложения с помощью «горячих клавиш» (*hotkeys*) или, например, для переключения неправильной раскладки клавиатуры (как Keyboard Ninja).

Существует масса легального ПО, которое используется администраторами для наблюдения за тем, что делает работник в течение дня, или для наблюдения пользователем за активностью посторонних людей на своем компьютере. Однако где проходит грань между «законным» использованием «легального» ПО и его использованием в криминальных целях? То же «легальное» ПО зачастую используется и в целях умышленного похищения секретных данных пользователя — например, паролей.

Большинство существующих на данный момент кейлоггеров считаются «легальными» и свободно продаются, так как разработчики декларируют множество причин для использования кейлоггеров, например:

- для родителей: отслеживание действий детей в Интернете и оповещение родителей в случае попыток зайти на сайты «для взрослых» (*parental control*);
- для службы безопасности организации: отслеживание фактов нецелевого использования персональных компьютеров, их использования в нерабочее время;

- для службы безопасности организации: отслеживание фактов набора на клавиатуре критичных слов и словосочетаний, которые составляют коммерческую тайну организации, и разглашение которых может привести к материальному или иному ущербу для организации;
- для различных служб безопасности: проведение анализа и расследования инцидентов, связанных с использованием персональных компьютеров;
- другие причины

В отличие от других типов вредоносного программного обеспечения, для системы кейлоггер абсолютно безопасен. Однако он может быть чрезвычайно опасным для пользователя: с помощью кейлоггера можно перехватить пароли и другую конфиденциальную информацию, вводимую пользователем с помощью клавиатуры. В результате злоумышленник узнает коды и номера счетов в электронных платежных системах, пароли к учетным записям в online-играх, адреса, логины, пароли к системам электронной почты и так далее.

После получения конфиденциальных данных пользователя злоумышленник может не только банально перевести деньги с его банковского счета или использовать учетную запись пользователя в online-игре. К сожалению, наличие таких данных в ряде случаев может приводить к последствиям более серьезным, чем потеря некоторой суммы денег конкретным человеком. Использование кейлоггеров позволяет осуществлять экономический и политический шпионаж, получать доступ к сведениям, составляющим не только коммерческую, но и государственную тайну, а также компрометировать системы безопасности, используемые коммерческими и государственными структурами (например, с помощью кражи закрытых ключей в криптографических системах).

Каждый вирус имеет собственное имя. Вы слышите его, когда узнаете об очередной эпидемии. Откуда берется имя? Обнаружив новый вирус, антивирусные компании дают ему имена в соответствии с классификациями, принятыми в каждой конкретной компании, причем классификация у каждой фирмы своя.

Посмотрите сами: например, Worm.Win32.Nuf – это то же самое, что Net-Worm.Win32.MytoB.c.

Часто название дается по некоторым внешним признакам:

- по месту обнаружения вируса (Jerusalem);
- содержащимся в теле вируса текстовым строкам (I Love You);
- методу подачи пользователю (AnnaKournikova);
- эффекту (Black Friday).

Антивирусные программы

Антивирусными программами называются специальные программы, которые позволяют обнаруживать и уничтожать вирусы.

Виды:

- программы-детекторы
- программы-доктора или фаги
- программы-ревизоры
- программы – фильтры
- программы-вакцины или иммунизаторы

Программы-доктора или фаги, а также программы-вакцины не только находят зараженные вирусами файлы, но и «лечат» их, т.е. удаляют из файла тело программы-вируса, возвращая файлы в исходное состояние. В начале своей работы фаги ищут вирусы в ОЗУ, уничтожая их, и только затем переходят к «лечению» файлов. Среди фагов выделяют полифаги, т.е. программы-доктора, предназначенные для поиска и уничтожения большого количества вирусов. Наиболее известные из них: Aidstest, Scan, Norton AntiVirus, Doctor Web.

Для того, чтобы не подвергать компьютер заражению вирусами и обеспечить надежное хранение информации на дисках, необходимо соблюдать следующие правила:

- оснастите свой компьютер современными антивирусными программами Doctor Web, и постоянно обновляйте их версии;
- перед считыванием информации записанных на другом компьютере, всегда проверяйте эти устройства на наличие вирусов
- при переносе на свой ПК файлов в архивированном виде проверяйте их сразу же после разархивации на жестком диске, ограничивая область проверки только вновь записанными файлами;
- периодически проверяйте жесткие диски ПК на наличие вирусов, запуская антивирусные программы для тестирования файлов, памяти и системных областей дисков;
- обязательно делайте архивные копии ценной для вас информации
- не оставлять в дисководы дискеты при включении или перезагрузке операционной системы, чтобы исключить заражение компьютера загрузочными вирусами;
- использовать антивирусные программы для входного контроля всех исполняемых файлов, получаемых из компьютерных сетей.

Тема 2.5. Интернет (лекция – дискуссия 2 часа)

Протоколы Интернет

1. Internet как иерархия сетей

Слово *Internet* происходит от выражения *interconnected networks* (связанные сети). Это глобальное сообщество малых и больших сетей. В широком смысле – это глобальное информационное пространство, хранящее огромное количество информации на миллионах компьютеров, которые обмениваются данными.

К концу 1969 г. в США был завершен проект ARPAnet подключением в одну компьютерную сеть 4 исследовательских центров: University of California Los Angeles, Stanford Research Institute, University of California at Santa Barbara, University of Utah. Проект также предусматривал проведение экспериментов в области компьютерных коммуникаций, изучение способов поддержания связи в условиях ядерного нападения и разработку концепции децентрализованного управления военными и гражданскими объектами в период ведения войн. В 1972 г. Минобороны США начало разработку новой программы Internetting Projects целью изучения методов соединения сетей между собой. Выдвигались требования максимальной надежности передачи данных при заведомо низком качестве коммуникаций, средств связи и оборудования и возможности передачи больших объемов информации. В 1974 г. была поставлена задача разработки универсального протокола передачи данных, которая была решена созданием протокола передачи данных и объединения сетей – Transmission Control Protocol/Internet Protocol (TCP/IP). В 1983 г. был осуществлен перевод ARPAnet на TCP/IP. В 1989 г. в Европейской лаборатории физики элементарных частиц (CERN, Швейцария, Женева) Тим Бернерс-Ли разработал технологию *гипертекстовых документов – World Wide Web*, позволяющую пользователям иметь доступ к любой информации, находящейся в сети Интернет на компьютерах по всему миру. К 1995 г. темпы роста сети показали, что регулирование вопросов подключения и финансирования не может находиться в руках одного Национального научного фонда США, и в этом же году произошла передача региональным сетям оплаты за подсоединение многочисленных частных сетей к национальной магистрали.

Рассмотрим схему подключения компьютера к Интернету и проследим, по каким каналам передается информация, посылаемая в Сеть и принимаемая из Сети. Подключение к Интернету домашнего компьютера выполняется, как правило, с помощью *модема* (рис. 1). При этом чаще всего осуществляется так называемое *сеансовое* соединение с провайдером по телефонной линии. Набирается один из телефонных номеров, предоставленных провайдером, для соединения с одним из его модемов. У провайдера имеется набор модемов, так называемый *модемный пул*. После того, как вы соединились с ISP (Internet Service Provider), ваш компьютер становится *частью сети* данного ISP. Каждый провайдер имеет свою магистральную линию или *backbone*.

Рис. 1. Схема подключения компьютера к Интернету

ISP-провайдеры имеют так называемые точки присутствия POP (Point of Presence), где происходит подключение локальных пользователей. Провайдер может иметь точки присутствия POP в нескольких городах. В каждом городе находятся аналогичные модемные пулы, на которые звонят локальные клиенты этого провайдера в данном городе. Провайдер обычно арендует волоконно-оптические линии у телефонной компании для соединения всех своих точек присутствия. Крупные коммуникационные компании имеют собственные высокопропускные каналы.

Пусть имеются опорные сети двух Интернет-провайдеров. Очевидно, что все клиенты провайдера А могут взаимодействовать между собой по собственной сети, а все клиенты провайдера В — по своей, но при отсутствии связи между сетями А и В клиенты разных провайде-

ров не могут связаться друг с другом. Для реализации такой услуги провайдеры А и В подключаются к так называемым точкам доступа NAP (NetworkAccessPoints) в разных городах, и трафик между двумя сетями течет черезNAP. Аналогично организуется подключение к другим магистральным сетям, в результате чего образуется объединение множества сетей высокого уровня. В Интернете действуют сотни крупных провайдеров, их магистральные сети связаны черезNAPв различных городах, и миллиарды байтов данных текут по разным сетям черезNAP-узлы.

В офисе компьютеры, скорее всего, подключены к локальной сети. В этом случае рассмотренная схема видоизменяется. Варианты подключения к провайдеру могут быть различными, хотя чаще всего это выделенная линия.

На сегодняшний день существует множество компаний, имеющих собственные опорные сети (*бэбжуны*), которые связываются с помощьюNAPс сетями других компаний по всему миру. Благодаря этому каждый, кто находится в Интернете, имеет доступ к любому его узлу, независимо от того, где он расположен территориально.

Скорость передачи информации на различных участках Интернета существенно различается. Магистральные линии — это высокоскоростные каналы, построенные на основе волоконно-оптических кабелей. Кабели обозначаются *ОС* (opticalcarrier), например ОС-3, ОС-12 или ОС-48. Так, линия ОС-3 может передавать 155 Мбит/с, а ОС-48 — 2488 Мбит/с (2,488 Гбит/с). Но максимальная скорость получения информации на домашний компьютер с модемным подключением не превышает 56 Кбит/с.

Как же происходит передача информации по всем этим многочисленным каналам? Доставка информации по нужному адресу выполняется с помощью маршрутизаторов, определяющих, по какому маршруту передавать информацию. *Маршрутизатор* — это устройство, которое работает с несколькими каналами, направляя в выбранный канал очередной блок данных. Выбор канала осуществляется по адресу, указанному в заголовке поступившего сообщения.

Таким образом, маршрутизатор выполняет две взаимосвязанные функции. Во-первых, он направляет информацию по свободным каналам, предотвращая закупорку узких мест в Сети; во-вторых, проверяет, что информация следует в нужном направлении. При объединении двух сетей маршрутизатор включается в обе сети, пропуская информацию из одной в другую. В некоторых случаях он осуществляет перевод данных из одного протокола в другой, при этом защищая сети от лишнего трафика. Эту функцию маршрутизаторов можно сравнить с работой службы ГИБДД, которая ведет наблюдение за автомобильным движением с вертолета и сообщает водителям оптимальный маршрут.

2. Протоколы Интернет

Различают два типа протоколов: базовые и прикладные. *Базовые* протоколы отвечают за физическую пересылку сообщений между компьютерами сетиInternet. Это протоколыTCPиIP. *Прикладными* называют протоколы более высокого уровня, они отвечают за функционирование специализированных служб. Например, протоколHTTPслужит для передачи гипертекстовых сообщений, протоколFTP— для передачи файлов,SMTP— для передачи электронной почты.

Набор протоколов разных уровней, работающих одновременно, называют *стеком протоколов*. Каждый нижележащий уровень стека протоколов имеет свою систему правил и предоставляет сервис вышележащим. Аналогично каждый протокол в стеке протоколов выполняет свою функцию, не заботясь о функциях протокола другого уровня.

На нижнем уровне используются два основных протокола: *IP* (InternetProtocol– протокол Интернет) и *TCP* (TransmissionControlProtocol– протокол управления передачей). Архитектура протоколов TCP/IP предназначена для объединения сетей. В их качестве могут выступать разные ЛВС (TokenRing,Ethernetи др.), различные национальные, региональные и глобальные сети. К этим сетям могут подключаться машины разных типов. Каждая из сетей работает в соответствии со своими принципами и типом связи. При этом каждая сеть может принять пакет информации и доставить его по указанному адресу. Таким образом, требуется, чтобы каждая сеть имела некий сквозной протокол для передачи сообщений между двумя внешними сетями.

Предположим, имеется некое послание, отправляемое по электронной почте. Передача почты осуществляется по прикладному протоколу *SMTP*, который опирается на протоколы TCP/IP. Согласно протоколуTCP, отправляемые данные разбиваются на небольшие пакеты фиксированной структуры и длины, маркируются таким образом, чтобы при получении данные можно было бы собрать в правильной последовательности.

Обычно длина одного пакета не превышает 1500 байт. Поэтому одно электронное письмо может состоять из нескольких сотен таких пакетов. Малая длина пакета не приводит к блокировке линий связи и не позволяет отдельным пользователям надолго захватывать канал связи.

К каждому полученному TCP-пакету протоколIPдобавляет информацию, по которой можно определить адреса отправителя и получателя. Это аналогично помещению адреса на конверт. Для каждого поступающего пакета маршрутизатор, через который проходит пакет, по даннымIP-адреса определяет, кому из ближайших соседей необходимо переслать данный пакет, чтобы он быстрее оказался у получателя, т.е. принимает решение об оптимальном пути следования очередного пакета. При этом географически самый короткий путь не всегда оказывается оптимальным (быстрый канал на другой континент может быть лучше медленного в соседний город). Очевидно, что скорость и пути прохождения разных пакетов могут быть различными. Взаимосвязанные пакеты данных могут передаваться различными путями. Возможно, что пакеты будут путешествовать через разные континенты с различной скоростью. При этом пакеты, отправленные позже, могут дойти раньше. Независимо от длины пути в результате конечного числа пересылокTCP-пакеты достигают адресата.

Наконец, TCP-модуль адресата собирает и распаковываетIP-конверты, затем распаковываетTCP-конверты и помещает данные в нужной последовательности. Если чего-либо не достает, он требует переслать этот пакет снова. Пакеты не только теряются, но и могут искажаться при передаче из-за наличия помех на линиях связи.TCPрешает и эту проблему. В конце концов, информация собирается в нужном порядке и полностью восстанавливается.

Таким образом, протокол IPосуществляет перемещение данных в сети, а протоколTCPобеспечивает надежную доставку данных, используя систему кодов, исправляющих ошибки. Причем два сетевых сервера могут одновременно передавать в обе стороны по одной линии множествоTCP-пакетов от различных клиентов.

Необходимо подчеркнуть основное различие передачи информации по телефонной сети и по Интернету. Телефонная система при звонке по телефону в другой регион или даже на другой континент устанавливает канал между вашим телефоном и тем, на который вы звоните. Канал может состоять из десятков участков разной физической природы – медные провода, волоконно-оптические линии, беспроводные участки, спутниковая связь и т.д. Эти участки неизменны на протяжении всего сеанса связи. Это означает, что линия между вами и тем, кому вы звоните, постоянна в течение всего разговора, поэтому повреждения на любом участке линии способны прервать ваш разговор. При этом выделенная вам часть сети для других уже недоступна. Речь идет о *сети с коммутацией каналов*. Интернет же является*сетью с коммутацией пакетов*. Процесс пересылки электронной почты принципиально иной.

Итак, Internet-данные в любой форме – электронное письмо,Web-страница или скачиваемый файл – путешествуют в виде группы пакетов. Каждый пакет посылается на место назначения по оптимальному из доступных путей. Поэтому даже если какой-то участок Интернет окажется нарушенным, то это не повлияет на доставку пакета, который будет направлен по альтернативному пути. Таким образом, во время доставки данных нет необходимости в фиксированной линии связи между двумя пользователями.*Принцип пакетной коммутации* обеспечивает основное преимуществоInternet— надежность. Сеть может распределять нагрузку по различным участкам за тысячные доли секунды. Если какой-то участок оборудования сети поврежден, пакет может обойти это место и пройти по другому пути, обеспечив доставку всего послания. Прототип Интернет — сетьARPAnet, разработанная по заказу Минобороны США, задумывалась именно как сеть, устойчивая к повреждениям (например, в случае военных действий), способная продолжать нормальное функционирование при выходе из строя любой ее части.

3. Адресация в Интернет

Каждому компьютеру, подключенному к Интернету, присваивается *идентификационный номер*, который называется*IP-адресом*.

При сеансовом подключении к Интернету IP-адрес выделяется компьютеру только на время этого сеанса. Присвоение адреса компьютеру на время сеанса связи называется*динамическим распределением IP-адресов*. Оно удобно для провайдера, поскольку один и тот жеIP-адрес в разные периоды времени может быть выделен разным пользователям. Таким образом, Интернет-провайдер должен иметь по одномуIP-адресу на каждый обслуживаемый им модем, а не на каждого клиента.

IP-адрес имеет формат xxx.xxx.xxx.xxx, где xxx – числа от 0 до 255. Рассмотрим типичныйIP-адрес: 193.27.61.137. Для облегчения запоминанияIP-адрес обычно выражают рядом чисел в десятичной системе счисления, разделенных точками. Но компьютеры хранят его в бинарной форме. Например, тот жеIP-адрес в двоичном коде будет выглядеть так: 11000001.00011011.00111101.10001001.

Четыре числа в IP-адресе называются *октетами*, поскольку в каждом из них при двоичном представлении имеется $8 = 2^3$. Комбинация четырех октетов дает восемь разрядов: 4^2 значений, т.е. примерно 4,3 млрд комбинаций, за исключением некоторых зарезервированных адресов.

Оклеты делят на две секции: *Net* и *Host*. *Net-секция* используется для того, чтобы определить сеть, к которой принадлежит компьютер. *Host*, который называется *узлом*, определяет конкретный компьютер в сети.

Подобная система используется и в обычной почте.

На ранней стадии своего развития Интернет состоял из небольшого количества компьютеров, объединенных модемами и телефонными линиями. Тогда пользователи могли установить соединение с компьютером, набрав цифровой адрес, например 163. 25. 51. 132. Это было удобно, пока компьютеров было мало. По мере увеличения их количества цифровые имена стали заменять текстовыми, потому что текстовое имя проще запомнить, чем цифровое. Возникла проблема автоматизации этого процесса, и в 1983 г. в Висконсинском университете США была создана так называемая *DNS-система* (DomainNameSystem), которая автоматически устанавливала соответствие между текстовыми именами и IP-адресами. Вместо чисел была предложена ставшая сегодня для нас привычной запись типа www.myname.gorod.ru.

Подобным же образом осуществляется сортировка обычной почты. Люди привыкли ориентироваться по географическим адресам, в то время как автомат на почте быстро сортирует почту по индексу.

Таким образом, при пересылке информации компьютеры используют цифровые адреса, люди — буквенные, а DNS-сервер служит своеобразным переводчиком.

4. Доменные имена

Когда происходит обращение на *Web* или посылается *e-mail*, то используется доменное имя. Например, адрес <http://www.microsoft.com> содержит доменное имя microsoft.com. Аналогичное *e-mail-адреса* adrecalcol@rambler.ru содержит доменное имя rambler.ru.

В доменной системе имен реализуется принцип назначения имен с определением ответственности за их подмножество соответствующих сетевых групп.

Каждая группа придерживается этого простого правила. Имена, которые она присваивает, единственны среди множества ее непосредственных подчиненных, поэтому никакие две системы, где бы они ни находились в Интернете, не смогут получить одинаковые имена. Так же уникальны адреса, указываемые на конвертах при доставке писем обычной почтой. Таким образом, адрес на основе географических и административных названий однозначно определяет точку назначения.

Домены имеют подобную иерархию. В именах домены отделяются друг от друга точками: addressx.msk.ru, addressy.spb.ru. В имени может быть различное количество доменов, но обычно их не больше пяти. По мере движения по доменам в имени слева направо, количество имен, входящих в соответствующую группу, возрастает.

Для перевода буквенного доменного имени в IP-адрес цифрового формата служат *DNS-серверы*.

В качестве примера рассмотрим адрес group.facult.univers.tula.ru.

Первым в имени стоит название рабочей машины — реального компьютера с IP-адресом. Это имя создано и поддерживается группой faculty. Группа входит в более крупное подразделение univers, далее следует домен tula — он определяет имена тульской части ru — российской.

Каждая страна имеет свой домен: *au* — Австралия, *be* — Бельгия и т.д. Это географические домены верхнего уровня.

Помимо географического признака используется организационный признак, в соответствии с которым существуют следующие доменные имена первого уровня:

- com* — коммерческие предприятия,
- edu* — образовательные учреждения,
- gov* — государственные учреждения,
- mil* — военные организации,
- net* — сетевые образования,
- org* — учреждения других организаций и сетевых ресурсов.

Внутри каждого доменного имени первого уровня находится целый ряд доменных имен второго уровня. Домен верхнего уровня располагается в имени правее, а домен нижнего уровня — левее.

Так, в адресе www.tsu.tula.ru домен верхнего уровня указывает на то, что адрес принадлежит российской части Интернет, tula — определяет город, следующий уровень — домен конкретного предприятия или учреждения (*tsu* — тульский государственный университет).

Лавинообразное подключение в сети Интернет обдарило проблему недостатка адресного пространства. В 1995 г. организация IETF (Internet Engineering Task Force — инженерные силы Интернет) опубликовала рекомендации по протоколу IP следующего поколения — *IP v.6* (сейчас *IPv.4*), которые предполагают постепенный переход с существующей 32-разрядной системы присвоения IP-адресов на 128-разрядную систему. Такая мера сулит увеличение адресного пространства в 296 раз, что позволит каждому жителю планеты иметь несколько адресов. Переход уже начался. Вместе с использованием новых оптоволоконных каналов для увеличения скорости в сотни и тысячи раз расширение адресного пространства даст возможность осуществить проект Интернет 2. Эта сеть в настоящее время развертывается в США для ряда университетов, школ, федеральных агентств и крупных компьютерных компаний.

Во время приема запроса на перевод доменного имени в IP-адрес DNS-сервер выполняет одно из следующих действий:

- отвечает на запрос, выдав IP-адрес, если знает IP-адрес запрашиваемого домена;
- взаимодействует с другим DNS-сервером для того, чтобы найти IP-адрес запрошенного имени, если он его не знает (такой запрос может проходить по цепочке DNS-серверов несколько раз);
- выдает сообщение: «Я не знаю IP-адреса домена, запрашиваемого вами, но вот IP-адреса DNS-сервера, который знает больше меня»;
- сообщает, что такой домен не существует.

Предположим, вы набрали адрес group.facult.univers.tula.ru, который имеет адрес в домене верхнего уровня COM. В простейшем варианте браузер контактирует с DNS-сервером для того, чтобы получить IP-адрес искомого компьютера, и DNS-сервер возвращает этот искомый IP-адрес.

Одна из причин надежной работы этой системы — ее избыточность. Существует множество DNS-серверов на каждом уровне, и поэтому если один из них не может дать ответ, то точно существует другой, на котором есть необходимая информация.

Система кэширования делает поиск более быстрым. DNS-сервер, однажды сделав запрос на корневой DNS и получив адрес нужного DNS-сервера, кэширует полученный IP-адрес. В следующий раз он уже не будет повторно обращаться с подобным запросом. Подобное кэширование происходит с каждым запросом, что постепенно оптимизирует скорость работы системы. Пользователям работа DNS-сервера не видна, однако эти серверы каждый день выполняют миллиарды запросов, обеспечивая работу миллионов пользователей.

5. Варианты доступа в Интернет

Провести соединение между ISP-провайдером и пользователями — задача не из простых. Обычно провайдер подключен к Интернет с помощью дорогостоящего оптоволоконного высокоскоростного канала. Один провайдер обслуживает множество клиентов, которые рассредоточены на большой территории. Технология, по которой осуществляется связь между абонентами и местной телекоммуникационной службой, т.е. провайдером, получила название *технологии последней мили*. Название это условное (обычно расстояние от абонента до провайдера не превышает 4 км).

Существует целый ряд технологий, позволяющих использовать имеющуюся инфраструктуру — телефонные линии, сети кабельного телевидения и т.д., — для осуществления доступа в Интернет.

Наиболее распространенный среди домашних пользователей в России способ доступа в Интернет — *доступ по коммутируемой телефонной линии с помощью модема*. Скорость доступа при таком способе подключения не более 56 Кбит/с, но такая скорость сегодня мало кого устраивает. Какие же альтернативные технологии позволяют получить более высокую скорость доступа в Интернет?

Обычный телефон использует лишь низкочастотный диапазон линии. Однако провод телефонной линии способен передавать гораздо больше данных, если использовать более широкую полосу (полоса пропускания обычной телефонной линии 3400 КГц). Поэтому телефон-

ную сеть, которая изначально предназначалась для передачи голосового сигнала, приспособили для высокоскоростной передачи цифровых данных.

DSL-технология (DigitalSubscriberLine– цифровая абонентская линия) позволяет использовать более широкую полосу пропускания для передачи данных без ущерба для использования телефонной линии по прямому назначению. Существует целое семейство технологий под общим названием *DSL*, где приставка *x* указывает на конкретную спецификацию семейства *DSL*. Эта технология весьма перспективна, она позволяет одновременно работать в Интернете и разговаривать по телефону. Скорость подключения по ней намного выше, чем при помощи обычного модема. *DSL* не требует прокладки новых проводов, так как использует уже имеющуюся телефонную линию.

Одним из основных преимуществ технологии *xDSL* является высокоскоростной доступ в Интернет. При работе в Интернет основной поток информации идет из сети к пользователю, а в сеть передается гораздо меньший объем данных. Действительно, при просмотре Web-страниц в ответ на небольшой запрос пользователь получает из Сети не только текст, но и изображения. Таким образом, информационный обмен является *асимметричным*.

ADSL (AsymmetricalDSL), или асимметричный *DSL*, позволяет передавать данные пользователю со скоростью, на порядок превышающую скорость от пользователя. При этом сигнал от пользователя в Сеть передается на более низких частотах, чем сигнал из Сети к пользователю. Теоретически при этом можно иметь канал с пропускной способностью 1 Мбит/с в прямом направлении (в Сеть) и 8 Мбит/с – в обратном. При этом одна и та же линия может использоваться для передачи голоса и цифровых данных. По сравнению с коммутируемым доступом *ADSL*-линия работает, как минимум, на два порядка быстрее. Высокая скорость позволяет комфортно работать с Web-сайтами с мультимедийной информацией, быстро перекачивать большие файлы и полноценно использовать интерактивные приложения.

Достоинства *ADSL*: легкость установки (используется уже имеющаяся телефонная линия), постоянный доступ в Интернет (пользователи *ADSL* не разделяют полосу пропускания с другими абонентами).

Недостаток *ADSL*: ограничения по дальности. Скорость передачи потока данных в обратном направлении существенно зависит от расстояния. Если при расстоянии 3 км можно получить скорость около 8 Мбит/с, то на расстоянии 5 км – только 1,5 Мбит/с.

На стороне пользователя компьютер подключается к *ADSL*-модему. Принцип действия *ADSL*-модема заключается в том, что диапазон частот в интервале 24-1100 КГц разбивается на 4 КГц полосы, на каждую из которых назначается виртуальный модем. Таким образом, каждый из этих 249 виртуальных модемов работает со своим диапазоном. *ADSL*-модем подключается к частотному разделителю. Частотный разделитель представляет собой фильтр низких частот, разделяющий низкочастотный сигнал обычной телефонной связи и высокочастотный *ADSL*-сигнал. Конструктивно *частотный разделитель*, или *сплиттер*, выполняется в виде блока, имеющего три гнезда: для подключения *ADSL*-модема, телефонного аппарата и линии. Частотный разделитель позволяет подключить к одной линии и компьютер, и телефон. Таким образом, по одной линии могут передаваться и цифровые компьютерные сигналы, и аналоговые сигналы телефонной связи.

На телефонной станции такой же частотный разделитель позволяет разделять низкочастотные и высокочастотные сигналы на другом конце абонентной линии. Голосовой аналоговый сигнал направляется в телефонную сеть общего пользования, а цифровой сигнал – на мультиплексор доступа *DSLAM*. На стороне провайдера сигнал от мультиплексора доступа *DSLAM* через модемный пул и сервер попадает в Интернет.

Мультиплексор доступа DSLAM (DigitalSubscriberLineAccessMultiplexer) – это устройство, установленное на телефонной станции, которое осуществляет подключение всех *DSL*-абонентов к одной высокоскоростной линии.

ADSL – весьма экономичная технология. Обычно такая линия обходится потребителю намного дешевле, чем выделенный канал аналогичной пропускной способности. По данной технологии может быть подключен не только отдельный компьютер, но и локальная сеть.

DSL-технология позволяет также использовать *широкополосный доступ*. Понятие «широкополосный доступ» означает, что канал предоставляет *расширенную полосу частот* для передачи информации. Высокая скорость передачи информации достигается благодаря тому, что с использованием широкой полосы частот информация может быть мультиплексирована и отправлена на нескольких различных частотах, позволяя, таким образом, передавать за единицу времени большее количество информации. Как известно, *мультиплексирование* – это передача нескольких сигналов по одному физическому каналу путем *разделения* его на подканалы. Говоря о частотном мультиплексировании, имеют в виду частотное разделение на подканалы.

Под термином *узкополосный доступ* обычно понимается канал, достаточный для передачи голоса. Скорость передачи по такому каналу не превышает 64 Кбит/с. Считается, что широкополосный доступ – это канал со скоростью передачи не менее 256 Кбит/с. Широкополосный доступ позволяет передавать в одном канале различные сигналы и одновременно пользоваться телефоном, телевизором и Интернетом.

Выделенная телефонная линия – это арендованная телефонная линия связи, соединяющая без коммутации двух абонентов. Наиболее распространенной технологией выделенной линии является технология *ISDN* (IntegratedServicesDigitalNetwork).

ISDN – это стандарт цифровой передачи. Основным компонентом любой *ISDN*-линии является *однонаправленный канал* или *B-канал* с пропускной способностью 64 Кбит/с. По этому каналу могут передаваться цифровые данные и, соответственно, оцифрованные видео- и аудиоданные. Для расширения полосы пропускания *B*-каналы группируются по два. В состав группы включается также *D-канал* (16 Кбит/с), управляющий передачей данных.

Передача информации может осуществляться по обычному медному проводу. Пользователи, которые устанавливают *ISDN*-адаптер вместо модема, могут получить доступ в Интернет со скоростью до 128 Кбит/с. *ISDN* требует установки адаптеров на обоих концах линии передачи. *ISDN*-канал обычно предоставляется телефонными станциями. По линии *ISDN* можно вести телефонные разговоры и одновременно передавать данные в Интернет.

Сеть **кабельного телевидения** первоначально была разработана как система для передачи аналогового видеосигнала в одном направлении – в сторону пользователя. Позднее были созданы так называемые *кабельные модемы*, которые кодируют и передают данные по кабелю таким образом, что это не мешает передаче телевизионного сигнала. Основным достоинством этой технологии является то, что используются уже имеющиеся сети кабельного телевидения. При доступе в Интернет по сетям кабельного телевидения обеспечивается высокая скорость передачи информации. Полосы пропускания телевизионного кабеля вполне достаточно для предоставления услуг *последней мили* при скоростях, сравнимых с теми, что предоставляют операторы *DSL*.

В отличие от *ADSL*, которая обеспечивает высокоскоростную передачу данных по одной телефонной линии, сети кабельного телевидения являются сетями коллективного пользования. Кабельные модемы получают услугу от общего источника информации. Рабочая полоса частот кабельного модема разделяется между всеми пользователями, подключенными к линии, и, следовательно, зависит от количества одновременно работающих пользователей. Обычно к одной модемной системе подключается несколько десятков абонентов. Чем больше клиентов одновременно посылают данные, тем меньше скорость их передачи. На практике скорость передачи данных от пользователей при применении кабельного модема часто меньше, чем при использовании *ADSL*.

Для организации связи между пользователем и опорной точкой радиосети провайдера используют *радиоканал* для высокоскоростного доступа в Интернет. С помощью этой технологии к Интернету можно подключить как индивидуальных пользователей, так и *ЛВС*. Для этого у абонента устанавливается *радиомодем*, который подключается к сетевой карте ПК или *к хабу/маршрутизатору* (в случае подключения *ЛВС*). Радиомодем соединен с направленной антенной, установленной на крыше здания. Антенна абонента направляется на базовую станцию провайдера. Связь между точкой входа в Интернет провайдера и абонентом осуществляется по радиоканалу.

С помощью данной технологии можно также объединить в сеть несколько филиалов компании без кабельного соединения. Для этого в каждом подразделении устанавливается абонентский комплект: направленная антенна и радиомодем. Провайдер обеспечивает связь между всеми точками доступа фирмы и правильную маршрутизацию данных.

Оборудование **беспроводных сетей** работает в диапазоне частот 2,4 ГГц. Сигналы такой частоты распространяются вдоль прямой линии, соединяющей антенны, поэтому радиоканал может быть организован при условии *прямой видимости* между абонентской антенной и антенной провайдера. На практике направленные антенны обеспечивают дальность связи до 30 км.

Преимущества радиоканала: быстрая установка, мобильность (нет кабеля), высокая скорость (несколько Мбит/с в зависимости от оборудования), затраты (первоначальные затраты на оборудование выше, чем в случае выделенной линии, но абонентская плата ниже).

В случае отсутствия телефонных станций и кабельного телевидения может помочь **спутниковый доступ** в Интернет. При этом скорость доступа на порядок выше, чем по обычному модему через коммутируемую телефонную линию, но несоизмеримо ниже *ADSL*-доступа.

Существует две разновидности организации высокоскоростного доступа в Интернет по спутниковому каналу: *симметричная* и *асимметричная*.

В случае *симметричного доступа* клиент осуществляет передачу запроса на спутник и прием данных со спутника. Подобное решение является достаточно дорогим, как по части клиентского оборудования, так и по стоимости абонентской платы.

В случае *асимметричного доступа* клиент осуществляет передачу запроса на получение требуемой информации по наземному каналу, а принимает информацию со спутника. Пользователь связывается с любым провайдером Интернета через обычный телефонный модем. Используя этот канал связи, он регистрируется на сервере провайдера, который обеспечивает асимметричный доступ в Интернет. После авторизации весь поток информации, поступающей в адрес пользователя через Интернет, направляется к нему не по обычной телефонной линии, а через спутниковый канал.

В последние годы активно разрабатываются технологии, направленные на использование *бытовой электрической сети* для доступа в Интернет. Одно из важнейших преимуществ бытовой электрической сети состоит в ее распространенности. Поэтому идея передачи информации по такой сети очень перспективна. Поскольку бытовая электрическая сеть первоначально не была предназначена для передачи информации, то это создает ряд технических трудностей. Электропроводка характеризуется высоким уровнем шумов, быстрым затуханием высокочастотного сигнала, а также изменением коммуникационных параметров в зависимости от текущей нагрузки.

Несмотря на технические трудности, сегодня уже имеются технологии, позволяющие использовать силовую кабельную инфраструктуру. В частности, компании Nor.webiUnitedUtilities разработали технологию *DPL* (Digital Power Line), позволяющую передавать голос и пакеты данных через простые электрические сети 120/220 В со скоростью до 1 Мбит/с.

Ожидается, что DPL-технология сможет дать новый импульс развитию средств передачи данных по линиям электропитания и сделает возможным прямой доступ в Интернет практически из любой точки земного шара по минимальной стоимости. Пока эта технология не получила широкого распространения, однако в ближайшем будущем можно ожидать существенных изменений на рынке провайдерских услуг и снижения расценок на доступ в Сеть, включая цены на коммутируемые и выделенные линии.

Если эта технология получит распространение, она сможет значительно изменить расстановку сил на рынке предоставления Internet-доступа. Технология будет способствовать и появлению новых *принципов проектирования силовых электрических сетей* с учетом как энергетических, так и коммуникационных требований.

6. Система адресации URL

Чтобы найти документ в сети Интернет, достаточно знать ссылку на него — так называемый универсальный указатель на ресурс *URL* (Uniform Resource Locator — унифицированный указатель ресурса), который указывает местонахождение каждого файла, хранящегося на компьютере, подключенном к Интернету.

Адрес URL является сетевым расширением понятия *полного имени ресурса*, например, файла или приложения и пути к нему в операционной системе. *BURL*, кроме имени файла и директории, где он находится, указывается *сетевое имя компьютера*, на котором этот ресурс расположен, *протокол доступа к ресурсу*, который можно использовать для обращения к нему.

Рассмотрим некоторые URL:

`http://www.abc.def.ru/kartinki/SLIDE.htm`

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

Вторая часть `www.abc.def.ru` указывает на доменное имя.

Третья часть `kartinki/SLIDE.htm` показывает программе-клиенту, где на данном сервере искать ресурс. В данном случае ресурсом является файл в формате `html`, а именно `SLIDE.htm`, который находится в папке `kartinki`.

Имена директорий, содержащиеся в URL — *виртуальные* и не имеют ничего общего с реальными именами каталогов компьютера, на котором выполняется Web-сервер, а являются их псевдонимами. Ни один владелец компьютера, на котором выполняется Web-сервер, не позволяет постороннему пользователю, обращающемуся к Web-серверу через Интернет, иметь доступ к реальной файловой системе этого компьютера.

При написании URL важно правильно указывать верхние и нижние регистры. Дело в том, что Web-серверы функционируют под управлением разных операционных систем, а в некоторых из них имена файлов и приложений являются регистро-чувствительными.

В общем случае формат URL имеет вид: (протокол доступа) [://<домен>: <порт>] [/<директория><имя ресурса> [<параметры запроса>]].

Первая часть URL соответствует используемому протоколу доступа, например `HTTP://` (протокол передачи гипертекста), `FTP://` (`File Transfer Protocol` — протокол передачи файлов) и т.д.

Вторая часть URL-адреса указывает доменное имя, а также может указывать *номер порта*. Любой сервер предоставляет сервис, используя нумерованные порты. При этом каждая служба имеет свой номер порта. Клиенты подключаются к сервису по уникальному IP-адресу и по конкретному номеру порта. Так, если на компьютере функционируют Web-сервер и FTP-сервер, то обычно Web-сервер будет доступен по порту 80, а FTP-сервер — по порту 21. Каждый из распространенных сервисов имеет свой стандартный номер порта: WWW — 80, FTP — 21, ECHO — 7, TELNET — 23, SMTP — 25, GOPHER — 70 и т.д. Если номер порта не указан, то по умолчанию предполагается 80. В рассмотренном выше примере номер порта указан не был, поэтому он будет определен по умолчанию в связи с именем используемого протокола, в данном случае — HTTP.

При этом следует учитывать, что если устанавливается свой Web-сервер, то его можно поместить на другой свободный номер порта, например 920. В этом случае, если имя машины, например, `aaa.bbb.com`, то подключиться к этому серверу можно по URL `http://aaa.bbb.com:920`.

Третья часть URL-адреса — путь доступа к файлу — аналогичен пути к файлу на клиентском компьютере. Если этот путь не указан, по умолчанию используется стандартный отклик, определяемый в настройках Web-сервера. В частности, стандартным откликом на HTTP-запрос для ряда Web-серверов служит вывод файла с именем `index.html`.

Форумы. Чаты. Социальные сети

Обычно пользователи идентифицируют Интернет со службой *WWW* (*World Wide Web* — *Всемирная паутина*). Но это далеко не так, ибо WWW — одна из многочисленных служб Интернета. По аналогии Интернет можно сравнить с системой транспортных магистралей, а виды сервисов Интернет — с различными службами доставки.

В число наиболее часто используемых служб Интернет входят электронная почта, WWW, служба новостей Интернет, передача файлов по протоколу FTP, терминальный доступ по протоколу Telnet и ряд других служб.

Электронная почта. Электронная почта возникла раньше, чем Интернет, однако она не только не устарела, но, напротив, является наиболее массовой службой Сети и постоянно приобретает новых пользователей. Электронное письмо, как и обычное, содержит *адреса отправителя и получателя*. В него можно вложить графическое изображение или иной файл — точно так же, как в конверт с письмом можно положить открытку или фотографию. На него можно поставить электронную подпись, которая играет ту же роль, что и подпись в обычном письме. Однако служба *e-mail* давно обошла по популярности традиционную почту: ежегодно в мире рассылается более 600 млрд. электронных писем. Чем же вызвана такая популярность? Для ответа на этот вопрос перечислим достоинства электронной почты.

В отличие от телефонного звонка электронная почта может быть прочитана в удобное время, что особенно важно с учетом разницы во времени между часовыми поясами. Следует также отметить демократичность электронной почты: отправляя электронное письмо хоть самому президенту, вы не рискуете отвлечь его от текущих дел. К удобствам электронной почты следует также отнести возможность рассылки писем сразу большому количеству получателей, высокую скорость доставки, удобство пересылки вложенных файлов. Хранение писем в базе данных почтового клиента позволяет осуществлять быстрый поиск и сортировку почтовых отправлений. Кроме того, электронная почта в несколько сот раз дешевле обычной почтовой рассылки.

Обычно в момент регистрации доступа в Интернет сервис-провайдер предоставляет пользователю дисковое пространство под почтовый ящик: адрес этого почтового ящика (`E-mailAccountAddress`), имя пользователя (`E-mailAccountLoginName`) и пароль (`E-mailAccountPassword`). Пароль для доступа предоставляется в целях предотвращения несанкционированного доступа к почте. Адрес электронной почты имеет формат: `имя_пользователя@имя_домена`, например `Ivanov@abc.rst.ru`.

Часть слева от значка `@` — это имя почтового ящика (`E-mailAccountName`) на сервере, из которого владелец этого адреса забирает письма (в данном примере — `Ivanov`). Как правило, имя пользователя совпадает с именем почтового ящика.

Часть справа от значка @ называется доменом и указывает на местонахождение этого почтового ящика. Нужно отметить, что носителем адреса электронной почты является вовсе не конечный пункт доставки, т.е. не адрес вашего домашнего компьютера, а адрес сервера, на котором вы будете получать почту.

Электронная почта построена по принципу *клиент-серверной* архитектуры. Пользователь общается с клиентской программой, которая, в свою очередь, общается с почтовым сервером. Очевидно, что процедуры отправки и получения почты требуют разной степени идентификации личности, поэтому существуют и два разных протокола – на отправку и на прием писем.

Для передачи писем используются протокол *SMTP* (SimpleMailTransferProtocol– простой протокол пересылки почты) и соответственно *SMTP-серверы*. Чаще всего отправка почты происходит с почтового сервера вашего Интернет-провайдера, хотя, в принципе, это не обязательно. Обычно SMTP-серверы не требуют идентификации, поэтому вы можете отправить письмо с любого такого сервера. Для приема почтовых сообщений и настоящее время наиболее часто используется протокол *POP3* (PostOfficeProtocol– протокол почтового офиса), который контролирует право пользователя забирать почту из ящика и поэтому требует предоставления имени пользователя и пароля.

Рассмотрим конкретный пример работы почты. Пусть некий владелец электронного ящика с адресом *petya@abc.ru* на почтовом сервере *abc.ru* пишет письмо владельцу почтового ящика с адресом *vasya@xyz.com* на сервере *xyz.com*.

Для того чтобы подготовить письмо, он вызывает клиентскую программу, создает текст сообщения и в графе *Кому* указывает адрес получателя *vasya@xyz.com*. Если отправитель не имеет постоянного подключения к Интернету, то после нажатия кнопки *Отправить* он устанавливает сеанс связи с провайдером и начинает получать накопившуюся почту и отправлять подготовленные письма. Порядок приема почты обычно зависит от текущих настроек почтовой программы. Нередко письма складываются в определенную папку и отправляются другой командой после установления связи с провайдером. Процесс загрузки на локальный компьютер вновь поступивших писем и отправления новых писем на сервер называется *синхронизацией учетной записи*.

После того как вы подключились к Интернету, клиентская программа соединяется с почтовым сервером и передает серверу почтовый адрес получателя *vasya@xyz.com* и текст самого сообщения. При отправке почты клиентская программа взаимодействует с сервером исходящей почты, т.е. с SMTP-сервером, по протоколу SMTP, подключаясь к порту с номером 25. Процедура отправки электронной почты заключается в копировании вновь подготовленных сообщений из базы клиента в базу почтового сервера (в нашем случае — сервера *abc.ru*).

Рассмотрим этот процесс более подробно. После того как письмо доставлено на сервер отправителя, SMTP-сервер последнего должен связаться с сервером получателя. Для этого ему необходимо знать IP-адрес SMTP-сервера, получающего почту для адресов из домена *xyz.com*. Чтобы узнать этот адрес, он обращается к DNS-серверу и задает ему вопрос: Каков IP-адрес SMTP-сервера, получающего почту для адресов из домена *xyz.com*?

DNS-сервер выдает IP-адрес, после чего SMTP-сервер *naabc.ru* может соединиться с SMTP-сервером *naxyz.com*. Если по какой-либо причине SMTP-сервер *naabc.ru* не может связаться с SMTP-сервером *naxyz.com*, то послание встает в очередь для отправки. Обычно через каждые 15 минут производится попытка переслать послание из очереди. Через четыре часа отправителю будет послано сообщение о проблеме, а спустя пять дней большинство серверов прекращают попытки и возвращают отправителю недоставленную почту, так что бесследно ваше письмо не исчезнет.

Как только SMTP-серверу *naabc.ru* удается связаться с SMTP-сервером *naxyz.com*, он передает послание. Сервер *xyz.com* определяет, существует ли пользователь с именем *vasya*, и передает послание в ящик *vasya*.

После того, как почта оказалась на сервере *xyz.com*, получатель имеет возможность скачать ее оттуда. При получении почты, накопившейся в вашем почтовом ящике, клиентская программа получателя взаимодействует с POP3-сервером по протоколу POP3. Поскольку при входе в Интернет вы уже сообщаете свой пароль и логин, то обычно при получении почты дополнительное введение этих данных не требуется (следует отметить, что в принципе пароль и доступ к ящику не всегда один и тот же). Ваш почтовый клиент связывается с POP3-сервером и передает команды, которые определяют передачу копий посланий электронной почты на локальную машину клиента.

WWW — самый популярный сервис Интернета. Именно он, благодаря своей относительной простоте и наглядности для пользователей, сделал столь массовыми обращения к ресурсам Сети.

В самом общем плане WWW — это система Web-серверов, поддерживающая документы, форматированные специальным образом. Служба WWW реализована в виде клиент-серверной архитектуры. Пользователь с помощью клиентской программы (браузера) осуществляет запрос той или иной информации на сервере, а Web-сервер обслуживает запрос браузера. Браузер — это программа, обеспечивающая обращение к искомому ресурсу на сервере по его URL, интерпретирующая полученный результат и демонстрирующая его на клиентском компьютере.

Протокол, по которому происходит доставка Web-сервером документа Web-браузеру, носит название HTTP (Hypertext Transfer Protocol – протокол передачи гипертекста). Гипертекст – это текст, содержащий гиперссылки, связывающие слова или картинки документа с другим ресурсом (с каким-нибудь еще документом или с иным разделом этого же документа), при этом подобные связанные слова или картинки документа, как правило, выделяют, обычно с помощью подчеркивания. Пользователь может активировать эту связь щелчком мыши. Поскольку современные электронные документы содержат не только текст, но и любую мультимедийную информацию (текст, графика, звук), в качестве ссылок стали использовать не только текстовые, но и графические объекты. Со временем понятие гипертекста было расширено до понятия гипермедиа. Гипермедиа — это метод организации мультимедийной информации на основе ссылок на разные типы данных.

Особенно продуктивной идея гипертекста оказалась применительно к объединению цифровой информации, распределенной на серверах во всем мире.

WWW — это глобальная гипертекстовая система, организованная на базе Интернет. WWW представляет собой механизм, при помощи которого связывается информация, доступная посредством многочисленных Web-серверов во всем мире. Web-сервер — это программа, которая умеет получать http-запросы и выполнять в соответствии с этими запросами определенные действия, например запускать приложения и генерировать документы.

Документ, доступный через Web, называют Web-страницей, а группы страниц, объединенные общей темой и навигационно — Web-узлами, или Web-сайтами. Один аппаратный Web-сервер может содержать несколько Web-сайтов, но возможна и обратная ситуация, когда огромный Web-сайт может поддерживаться группой Web-серверов. Тот факт, что навигация не требует знаний о местоположении искомых документов, как раз и является основным удобством и причиной популярности службы WWW.

В браузерах реализованы две основные функции: запрос информации у Web-сервера и отображение ее на клиентском компьютере. Кроме того, браузеры обладают дополнительными сервисными функциями, такими как упрощение поиска, хранение закладок, указывающих на избранные страницы, и др.

Популярность WWW обусловлена тем, что можно не только просматривать чужие страницы и иметь доступ к огромному количеству информации, представленной на сотнях миллионов компьютеров, но и создать собственные ресурсы и таким образом донести любую информацию до всех будущих посетителей сайта. Иными словами, WWW — это глобальный механизм обмена информацией: одни люди помещают информацию на Web-серверы, а другие ее просматривают. Создав Web-сайт, владелец может поместить туда информацию различного рода: текст, графику, звук, анимацию, которая станет доступной для всех посетителей этого ресурса. С появлением и Сети вашей страницы информация о вас или о вашей фирме станет доступна сотням миллионов пользователей круглосуточно семь дней в неделю. Количество информации, которое может быть предоставлено посетителю, практически не ограничено по времени, в отличие от радио или телевидения.

8. Поиск в Интернете

Бытует мнение, что в Интернете есть все, но найти там что-либо практически невозможно. Впрочем, противоположная точка зрения, взятая на вооружение поисковой системой Яндекс, гласит, что найти в Интернете можно все. Видимо, для того чтобы находить, нужно уметь искать.

Для поиска в Интернете предназначены различные инструменты: поисковые машины, индексированные каталоги, метапоисковые системы, тематические списки ссылок, онлайн-ные энциклопедии и справочники. При этом для поиска разного рода информации наиболее эффективными оказываются различные инструменты. Рассмотрим каждый инструмент в отдельности.

Индексированные каталоги содержат информацию, иерархически структурированную по темам. Тематические разделы первого уровня определяют широко популярные темы, такие как спорт, отдых, наука, магазины и т.д. В каждом разделе есть подраздел. Таким образом, путешествуя по дереву каталога, можно постепенно сужать область поиска. Дойдя до нужного подкаталога, вы находите в нем набор ссылок. Обычно в каталоге все ссылки являются профильными, поскольку составлением каталогов занимаются не программы, а люди. Очевидно, что если вы ищете информацию по некоторой широкой теме, то целесообразно обратиться к каталогу. Если же вам необходимо найти конкретный документ, то каталог окажется малоэффективным поисковым средством. Один из наиболее популярных каталогов в России – List.ru находится по адресу <http://mail.ru/>. Кроме каталогов общего назначения в Сети много специализированных каталогов. Если внутри отдельной темы каталога находится огромное количество ресурсов, возникает проблема выбора. В некоторых каталогах имеется сортировка по популярности, например в каталоге Google. Яндекс сортировка идет по индексу цитирования.

Тематические списки ссылок – это списки, составленные группой профессионалов или коллекционерами-одиночками. Часто узкоспециализированная тема может быть раскрыта одним специалистом лучше, чем группой сотрудников крупного каталога. Тематических коллекций в Сети очень много, поэтому давать конкретные адреса не имеет смысла.

Поисковые машины. В ответ на запрос мы обычно получаем длинный список документов, многие из которых не имеют никакого отношения к теме запроса. Такие документы называются *нерелевантными*, т.е. не относящимися к делу. Таким образом, *релевантный* документ — это документ, содержащий искомую информацию. Очевидно, что от умения грамотно делать запрос зависит процент получаемых релевантных документов. Доля

релевантных документов в списке всех найденных поисковой машиной документов называется точностью поиска. Если все найденные документы релевантные, то точность поиска составляет 100%. Если найдены все релевантные документы, то полнота поиска – 100 %. Таким образом, качество поиска определяется двумя параметрами: точностью и полнотой поиска. Эти величины взаимосвязаны, т.е. увеличение полноты снижает точность, и наоборот.

Поисковая машина состоит из двух частей: *робота*, или паука, *и поискового механизма*. База данных робота формируется в основном им самим (робот сам находит ссылки на новые ресурсы) и в существенно меньшей степени – владельцами ресурсов, которые регистрируют свои сайты в поисковой машине. Помимо робота, который обходит все предписанные серверы и формирует базу данных, существует программа, определяющая рейтинг найденных ссылок.

Принцип работы поисковой машины сводится к тому, что она опрашивает свою базу данных по ключевым словам, которые пользователь указывает в поле запроса, и выдает список ссылок, ранжированный по релевантности.

Поиск по индексу заключается в том, что пользователь формирует запрос и передает его поисковой машине. В случае, когда у пользователя имеется несколько ключевых слов, весьма полезно использование булевых операторов. Текст, в пределах которого проверяется логическая комбинация, называется *единицей поиска*. Это может быть предложение, абзац или весь документ. В разных поисковых системах могут использоваться различные единицы поиска. После того, как пользователь сделал запрос, поисковая система обрабатывает синтаксис запроса и сравнивает ключевые слова со словами в индексе. После этого составляется список сайтов, отвечающих запросу, они ранжируются по релевантности, и формируется результат поиска, который и выдается пользователю.

Существует огромное количество поисковых систем. Наиболее популярная на Западе поисковая система – Google (www.google.com). Всемирно популярный каталог Yahoo! в качестве поисковой системы использует именно Google. В Рунете самыми популярными поисковыми системами являются Яндекс (www.yandex.ru), Google (www.google.ru) и Рамблер (www.rambler.ru).

Метапоисковые системы. Так как Интернет развивается стремительными темпами, то рост количества документов происходит быстрее, чем поисковые системы успевают их проиндексировать. Отсюда следует, что даже если в Сети и есть то, что вы ищете, вовсе не обязательно, что об этом знает та поисковая машина, к которой вы обратились. Велика вероятность, что нужный документ проиндексирован другой поисковой системой. Поэтому существуют службы, позволяющие транслировать запрос сразу в несколько поисковых систем, — это метапоисковые системы. Однако пользоваться ими во всех случаях не следует. Если документов по теме много, то метапоиск, возможно, даже вреден, поскольку смешивает разные логики ранжирования. Но если документов по теме мало, то метапоиск может быть полезен именно потому, что объединяет большое число поисковых систем.

Очень удобной в этом отношении является отечественная программа ДИСКo Искатель (www.disco.ru).

Онлайновые энциклопедии и справочники. Очень часто нужно найти не документ, содержащий то или иное ключевое слово, а именно – толкование искомого слова. Одной из крупнейших онлайн-энциклопедий является ресурс Яндекс Энциклопедии (<http://encycl.yandex.ru/>). Этот проект содержит 219 968 статей из 14 энциклопедий, в том числе из БСЭ и Энциклопедии Брокгауза и Ефрона. К крупным относится и Энциклопедия Кирилла и Мефодия, которую можно найти по адресу www.km.ru.

Особенно актуальным является поиск толкований терминов по информационным технологиям, которые развиваются так быстро, что уследить за появлением новых терминов очень сложно. Единственный ресурс на русском языке, который можно назвать компьютерным энциклопедическим словарем, — это проект *Компьютерная энциклопедия Кирилла и Мефодия* (<http://www.mega.km.ru/pc/>), предусматривающая поиск не только по термину, но и по тематической структуре. Объем словаря терминов — 700 статей. Объем англоязычного словаря *FOLDDOC* (FreeOnlineDictionaryOfComputing; <http://wombat.doc.ic.ac.uk/>) — более 13 тыс. терминов.

4.3. Лабораторные работы

<i>№ п/п</i>	<i>Номер раздела дисциплины</i>	<i>Наименование лабораторной работы</i>	<i>Объём (час.)</i>	<i>Вид занятия в ин- терактивной, активной, инновационной фор- мах, (час.)</i>
1.	1	Оператор присваивания	2	-
2.	1	Оператор условия	4	-
3.	1	Циклы в Qbasic	4	-
4.	1	Одномерный массив	4	-
5.	1	Двумерный массив	4	-
6.	1	Графика в Qbasic	4	-
7.	1	Построение графиков функции	4	-
8.	1	Вычисление площади криволинейной трапеции	4	-
9.	1	Подпрограммы	4	-
10.	2	Текстовый редактор Microsoft Word	8	-
11.	2	Табличный процессор Microsoft Excel	9	-
ИТОГО			51	-

4.4. Семинары / практические занятия

Учебным планом не предусмотрено.

4.5. Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат

Учебным планом не предусмотрено.

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

<i>№, наименование разделов дисциплины</i>	<i>Компетенции</i>	<i>Кол-во часов</i>	<i>Компетенции</i>		<i>Σ комп.</i>	<i>t_{ср}, час</i>	<i>Вид учебных занятий</i>	<i>Оценка результатов</i>
			<i>ОПК</i>	<i>ОПК</i>				
			<i>2</i>	<i>3</i>				
1. Теоретическая информатика. Основы программирования.		72	+	+	2	36,0	Лк, ЛР, СР	Зачет, Экзамен
2. Программное обеспечение ЭВМ. Интернет		81	+	+	2	40,5	Лк, ЛР, СР	Экзамен
<i>всего часов</i>		153	76,5	76,5	2	76,5		

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

1. Информатика: лабораторный практикум / Сост.: О.В. Вельц, И.П. Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL:

https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

Лабораторная работа № 1. Стр. 144 – 151.

Лабораторная работа № 2. Стр. 151 – 158.

Лабораторная работа № 3. Стр. 163 – 167.

Лабораторная работа № 4. Стр. 167 – 172.

Лабораторная работа № 5. Стр. 172 – 177.

Лабораторная работа № 6. Стр. 122 – 129.

Лабораторная работа № 7. Стр. 122 – 129.

Лабораторная работа № 10. Стр. 34 – 59.

Лабораторная работа № 11. Стр. 86 – 105.

2. Грошев А.С. Информатика. Лабораторный практикум / А.С. Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: с.

https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Лабораторная работа № 1. Стр. 116 – 124.

Лабораторная работа № 2. Стр. 124 – 134.

Лабораторная работа № 3. Стр. 134 – 141.

Лабораторная работа № 8. Стр. 146 – 150.

Лабораторная работа № 9. Стр. 146 – 150.

Лабораторная работа № 10. Стр. 15 – 41.

Лабораторная работа № 11. Стр. 41 – 84.

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

№ п/п	Наименование издания	Вид занятия	Кол-во экз. в библи., шт.	Обеспеченность, (экз./ чел.)
Основная литература				
1.	Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538	Лк, ЛР, СР	ЭР	1
2.	Информатика I: учебное пособие / И.Л.Артёмов, А.В.Гураков, О.И.Мещерякова, Д.С.Шкльц. – Томск: ФДО, ТУСУР, 2015. – 234 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=480593	Лк, ЛР, СР	ЭР	1
3.	Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784	Лк, ЛР, СР	ЭР	1
Дополнительная литература				
4.	Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915	Лк, ЛР, СР	ЭР	1
5.	Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590	Лк, ЛР, СР	ЭР	1

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=.
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

№ п/п	Номер раздела Основные положения раздела, рекомендуемые для СР	Рекомендуемая литература	Форма отчета	Всего часов
1.	1. Теоретическая информатика. Основы программирования. 1.1 Информатика и информация. 1.2. Структура ЭВМ. Поколения ЭВМ. 1.3. Алгоритм 1.4. Язык программирования QBASIC	[1], [2], [3], [4]	Зачет, Экзамен, ЛР № 1...9	21
2.	2. 2. Программное обеспечение ЭВМ. Интернет. 2.1. Операционные системы. 2.2. Текстовый редактор. 2.3. Табличный процессор. 2.4. Архиваторы. Антивирусы. 2.5. Интернет.	[1], [2], [3], [4]	Экзамен, ЛР № 10, 11	47
ИТОГО				68

9.1. Методические указания для обучающихся по выполнению лабораторных работ

Лабораторная работа №1 Оператор присваивания

Цель работы:

Изучение приёмов работы с оператором присваивания и построения блок-схем.

Содержание работы

Освоение приёмов работы с оператором присваивания в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Нарисовать блок-схему в текстовом редакторе Microsoft Word.
5. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic и блок-схему программы выполненную в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Информатика I: учебное пособие / И.Л.Артёмов, А.В.Гураков, О.И.Мещерякова, Д.С.Шкльц. – Томск: ФДО, ТУСУР, 2015. – 234 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=480593

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Оператор присваивания в Qbasic?
2. Как оператор присваивания изображается на блок-схеме?

Лабораторная работа №2 Оператор условия

Цель работы:

Изучение приёмов работы с оператором условия и построения блок-схем.

Содержание работы

Освоение приёмов работы с оператором условия в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Нарисовать блок-схему в текстовом редакторе Microsoft Word.
5. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic и блок-схему программы выполненную в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Информатика I: учебное пособие / И.Л.Артёмов, А.В.Гураков, О.И.Мещерякова, Д.С.Шкльц. – Томск: ФДО, ТУСУР, 2015. – 234 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=480593

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Оператор условия в Qbasic?
2. Как условие изображается на блок-схеме?

Лабораторная работа №3 Циклы в Qbasic

Цель работы:

Изучение приёмов работы с операторами циклов и построения блок-схем.

Содержание работы

Освоение приёмов работы с операторами циклов в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Нарисовать блок-схему в текстовом редакторе Microsoft Word.
5. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic и блок-схему программы выполненную в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Информатика I: учебное пособие / И.Л.Артёмов, А.В.Гураков, О.И.Мещерякова, Д.С.Шкльц. – Томск: ФДО, ТУСУР, 2015. – 234 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=480593

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Что такое цикл?
2. Какие бывают циклы?
3. Как цикл изображается на блок-схеме?

Лабораторная работа №4 Одномерный массив

Цель работы:

Изучение приёмов работы с массивами данных.

Содержание работы

Освоение приёмов работы с одномерным массивом в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Нарисовать блок-схему в текстовом редакторе Microsoft Word.
5. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic и блок-схему программы выполненную в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. - URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Что такое одномерный массив?
2. Какие операции можно производить с одномерным массивом?

Лабораторная работа №5 Двумерный массив

Цель работы:

Изучение приёмов работы с массивами данных.

Содержание работы

Освоение приёмов работы с двумерным массивом в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Нарисовать блок-схему в текстовом редакторе Microsoft Word.
5. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic и блок-схему программы выполненную в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. - URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Что такое двумерный массив?
2. Какие операции можно производить с двумерным массивом?

Лабораторная работа №6 Графика в Qbasic

Цель работы:

Изучение приёмов работы с графикой в Qbasic.

Содержание работы

Освоение приёмов работы с графикой в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. - URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Что такое режим экрана?
2. Какие графические примитивы есть в Qbasic?

Лабораторная работа №7 **Построение графиков функции**

Цель работы:

Изучение приёмов построения графиков функции.

Содержание работы

Освоение приёмов построения графиков функций в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic.

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. - URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Что такое область определения функции?
2. Перевод из полярных координат в декартовые?

Лабораторная работа №8 **Вычисление площади криволинейной трапеции**

Цель работы:

Изучение методов вычисления площади криволинейной трапеции.

Содержание работы

Освоение численных методов, используемых для вычисления площади криволинейной трапеции в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Нарисовать блок-схему в текстовом редакторе Microsoft Word.
5. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic и блок-схему программы выполненную в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 – 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. - URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 – 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. – 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Какие существуют численные методы вычисления площади криволинейной трапеции?
2. Как определить площадь криволинейной трапеции методом Монте-Карло?

Лабораторная работа №9 **Подпрограммы**

Цель работы:

Изучение приёмов работы с подпрограммами.

Содержание работы

Освоение приёмов работы с подпрограммами в Qbasic.

Порядок выполнения работы

1. Изучить необходимые команды.
2. Получить индивидуальное задание.
3. По выданному заданию написать программу на языке программирования Qbasic.
4. Нарисовать блок-схему в текстовом редакторе Microsoft Word.
5. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: программу Qbasic и блок-схему программы выполненную в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 - 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. - URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 - 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. - 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. что такое подпрограмма?

2. Какие подпрограммы используются в Qbasic?

Лабораторная работа №10 Текстовый редактор Microsoft Word

Цель работы:

Изучение текстового редактора Microsoft Word.

Содержание работы

Освоение приёмов работы в текстовом редакторе Microsoft Word.

Порядок выполнения работы

1. Изучить теоретические основы.

2. Получить индивидуальное задание.

3. По выданному заданию создать документ в Microsoft Word.

4. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: документ в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 - 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. - URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В.Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 - 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. - 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Что такое текстовый редактор?

2. Какие основные функции текстового редактора?

Лабораторная работа №11 Табличный процессор Microsoft Excel

Цель работы:

Изучение табличного процессора Microsoft Excel.

Содержание работы

Освоение приёмов работы в табличном процессоре Microsoft Excel.

Порядок выполнения работы

1. Изучить теоретические основы.

2. Получить индивидуальное задание.

3. По выданному заданию создать документ в Microsoft Excel.

4. Проанализировать полученные результаты.

Форма отчётности: отчёт по лабораторной работе должен содержать: документ в Microsoft Word

Рекомендации по выполнению заданий и подготовке к лабораторной работе

Лабораторная работа выполняется в соответствии с информацией, полученной на лекционном курсе, а также собранной студентом самостоятельно из предложенных источников. Полученные результаты обсуждаются и согласовываются с ведущим преподавателем.

Основная литература

1. Информатика и программирование: учеб. пособие / Р.Ю.Царев, А.Н.Пупков, В.В.Самарин, Л.А.Свечникова, Е.В.Мыльникова. - Красноярск: Сиб. федер. ун-т, 2014 - 132 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=364538

2. Платонов Ю.М., Уткин Ю.Г., Иванов М.И. Информатика. Учебное пособие. М. Альтаир - МГАВТ 2014, - 226 стр. [Электронный ресурс]. - URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=429784

Дополнительная литература

3. Информатика: лабораторный практикум / Сост.: О.В. Вельц, И.П.Хвостова. - Ставрополь: Изд-во СКФУ, 2017 - 196 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=466915

4. Грошев А.С. Информатика. Лабораторный практикум / А.С.Грошев. М.-Берлин: Директ-Медиа, 2015. - 159 с. [Электронный ресурс]. URL: https://biblioclub.ru/index.php?page=book_view_red&book_id=428590

Контрольные вопросы для самопроверки

1. Что такое табличный процессор?

2. Какие основные функции табличного процессора?

9.2. Методические указания по выполнению курсового проекта (курсовой работы), контрольной работы, РГР, реферата

Учебным планом не предусмотрено.

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Информационно-коммуникативные технологии (ИКТ) используются для:

- получения информации при подготовке к занятиям;
- создания презентационного сопровождения занятий;
- работы в электронной информационной среде.

Стандартное лицензионное программное обеспечение:

1. Microsoft Imagine Premium: Microsoft Windows Professional 7.
2. Microsoft Office 2007 Russian Academic OPEN No Level.
3. Антивирусное программное обеспечение Kaspersky Security.
4. Visual Basic 5.0 (Copyright © 1987-1999 Microsoft Corp.)
5. 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>№ Лк, ЛР, ПЗ</i>
Лк	Лекционная аудитория (мультимедийный класс)	Учебная мебель; Проектор мультимедийный «CASIO» XJ-UT310WN с настенным креплением CASIOYM-88 Интерактивная доска Promethean 88 Activ-BoardTouchDryErase 6 касаний с настенным креплением и программным обеспечением PrometheanActivInspire Монитор 17"LG L1753-SF (silver-blek) Системный блок (AMD 690G,mANX,HDD Seagate 250Gb,DIMM DR//2*512Mb, DVDRV, FDD	Лк 1.1...2.5
ЛР	Лаборатория автоматизации систем проектирования	Учебная мебель; Системный блок (AMD 690G, mANX, HDDSeagate 250Gb, DIMMDDR//2*512Mb, DVDRV, FDD; Системный блок CelD-315; Системный блок CPU 4000.2*512MB; Монитор Терминал TFT 19 LG L1953S-SF; Системный блок AMD Athlon 64X2; Системный блок Celeron 2,66; Сканер HP 3770; Монитор 15 LG; Системный блок iCel 433; Принтер HP LJ P2015	ЛР № 1...11
СР	Читальный зал № 1	Учебная мебель; 10-ПК i5-2500/H67/4Gb (монитор TFT19 Samsung); принтер HP LaserJet P2055D.	-

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

1. Описание фонда оценочных средств (паспорт)

№ компетенции	Элемент компетенции	Раздел	Тема	ФОС
ОПК-2	- 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. Теоретическая информатика. Основы программирования. 2. Программное обеспечение ЭВМ. Интернет.	1.1. Информатика и информация. 1.2. Структура ЭВМ. Поколения ЭВМ. 1.3. Алгоритм 1.4. Язык программирования QBASIC.	Экзаменационные вопросы, вопросы к зачёту
ОПК-3	- способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.		1.1. Информатика и информация. 1.2. Структура ЭВМ. Поколения ЭВМ. 1.3. Алгоритм 1.4. Язык программирования QBASIC 2.1. Операционные системы. 2.2. Текстовый редактор. 2.3. Табличный процессор. 2.4. Архиваторы. Антивирусы. Интернет.	Экзаменационные вопросы

2. Экзаменационные вопросы, вопросы к зачету

№ п/п	Компетенции		ЭКЗАМЕНАЦИОННЫЕ ВОПРОСЫ	№ и наименование раздела
	Код	Определение		
1.	ОПК-2	- 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. Информатика и информация. 2. Структура ЭВМ. 3. Поколения ЭВМ. 4. Алгоритм 5. Языки программирования 6. Операционные системы. 7. Текстовые редакторы. 8. Табличный процессор. 9. Архиваторы. Антивирусы. 10. Интернет.	1. Теоретическая информатика. Основы программирования. 2. Программное обеспечение ЭВМ. Интернет.
2.	ОПК-3	- способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.		

№ п/п	Компетенции		ВОПРОСЫ К ЗАЧЕТУ	№ и наименование раздела
	Код	Определение		
1.	ОПК-2	- 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. Информатика и информация. 2. Структура ЭВМ. 3. Поколения ЭВМ. 4. Алгоритм 5. Язык программирования QBASIC.	1. Теоретическая информатика. Основы программирования.
2.	ОПК-3	- способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.		

3. Описание показателей и критериев оценивания компетенций

Показатели	Оценка	Критерии
<p>знать: <i>ОПК-2</i></p> <p>- основы информационной и библиографической культуры, основные информационно-коммуникационные технологии и основные требования информационной безопасности;</p> <p><i>ОПК-3</i></p> <p>- современные информационные технологии, прикладные программные средства используемые при решении задач профессиональной деятельности;</p> <p>уметь: <i>ОПК-2</i></p> <p>- решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учётом основных требований информационной безопасности;</p> <p><i>ОПК-3</i></p> <p>- использовать современные информационные технологии, прикладные программные средства используемые при решении задач профессиональной деятельности;</p> <p>Владеть: <i>ОПК-2</i></p> <p>- способами применения информационно-коммуникационных технологий для решения стандартных профессиональных задач.</p> <p><i>ОПК-3</i></p> <p>- способностью использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.</p>	отлично	<ul style="list-style-type: none"> - даны исчерпывающие и обоснованные ответы на все поставленные вопросы; - ответы изложены грамотно, уверенно, логично, последовательно; - опираясь на усвоенные знания, четко увязывает научные положения с практической деятельностью; - свободно владеет основными понятиями дисциплины.
	хорошо	<ul style="list-style-type: none"> - даны полные, достаточно обоснованные ответы на поставленные вопросы; - твердо усвоил тему, грамотно и по существу излагает ее, опираясь на полученные знания; - не допускает существенных неточностей; - увязывает усвоенные знания с практической деятельностью; - владеет системой основных понятий дисциплины.
	удовлетворительно	<ul style="list-style-type: none"> - даны в основном правильные ответы на все поставленные вопросы, но без достаточного обоснования; - допускает несущественные ошибки и неточности; - испытывает затруднения в практическом применении полученных знаний; - слабо аргументирует научные положения; - частично владеет системой основных понятий дисциплины.
	неудовлетворительно	<ul style="list-style-type: none"> - допускает существенные ошибки и неточности при ответе на поставленные вопросы; - испытывает трудности в практическом применении полученных знаний; - не может аргументировать научные положения; - не владеет системой основных понятий дисциплины.
	зачтено	<ul style="list-style-type: none"> - даны исчерпывающие и обоснованные ответы на все поставленные вопросы; - ответы изложены грамотно, уверенно, логично, последовательно; - опираясь на усвоенные знания, четко увязывает научные положения с практической деятельностью; - свободно владеет основными понятиями дисциплины.
	не зачтено	<ul style="list-style-type: none"> - допускает существенные ошибки и неточности при ответе на поставленные вопросы; - испытывает трудности в практическом применении полученных знаний; - не может аргументировать научные положения; - не владеет системой основных понятий дисциплины.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

Дисциплина «Информатика» направлена на способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности. Процесс прохождения дисциплины включает изучение методов обработки информации, отечественного и зарубежного прикладного программного обеспечения.

Изучение дисциплины «Основы технологии машиностроения» предусматривает:

- лекции;
- лабораторные работы;
- самостоятельную работу;
- зачет;
- экзамен.

В ходе освоения раздела 1 «Теоретическая информатика. Основы программирования» студенты должны изучить основные понятия дисциплины информатика, познакомиться со структурой ЭВМ, познакомиться с алгоритмами, освоить основы программирования в среде Qbasic.

В ходе освоения раздела 2 «Программное обеспечение ЭВМ. Интернет» студенты должны изучить функции операционных систем, познакомиться с программным обеспечением ЭВМ, изучить основы работы в сети Интернет.

Необходимо овладеть умениями сбора информации и навыками работы в прикладном программном обеспечении.

В процессе изучения дисциплины рекомендуется обратить внимание на прикладное программное обеспечение, которое используется технологами для работы.

Овладение навыками работы с прикладным программным обеспечением является необходимым для создания комплекта технической документации при оформлении выпускной квалификационной работы.

При подготовке к зачёту, экзамену рекомендуется особое внимание уделить вопросам автоматизации обработки информации с использованием пакета прикладных программ Microsoft Imagine Premium.

В процессе проведения лабораторных работ, происходит закрепление знаний о основах программирования, приобретение умений решения технических задач с использованием необходимого прикладного программного обеспечения.

Самостоятельную работу необходимо начинать с ознакомления с предложенной основной и дополнительной литературой для последующего рассмотрения вопросов, связанных с информационными процессами.

Предусмотрено проведение аудиторных занятий в интерактивной, активной, инновационной формах с дискуссией в сочетании с внеаудиторной работой.

АННОТАЦИЯ
рабочей программы дисциплины
Информатика

1. Цель и задачи дисциплины

Целью изучения дисциплины является:

- сформировать знания, умения и навыки, необходимые для рационального использования средств современных информационно-коммуникационных технологий при решении задач, связанных с обработкой информации, ее поиском, систематизацией, сохранением, представлением, передачей;
- ознакомить студентов с ролью новых информационно-коммуникационных технологий в современном производстве, науке, повседневной практике, с перспективами развития компьютерной техники;

Задачами изучения дисциплины являются:

- получение представления об информатике (предмет, задачи, методы и сущность информатики) и освоение базовых понятий о средствах современной вычислительной техники и принципы работы технических и программных средств;
- ознакомление с алгоритмами и языком программирования, как средством реализации алгоритма на ЭВМ;
- изучение основных принципов программирования;
- ознакомление с пакетами прикладных программ, применяемых для обработки данных
- приобретение навыков в использование современной компьютерной техники для сбора, хранения и обработки информации;
- приобретение навыков работы на ЭВМ.

2. Структура дисциплины

2.1. Распределение трудоёмкости по отдельным видам учебных занятий, включая самостоятельную работу: лекции – 34 часа; лабораторные работы – 51 час; самостоятельная работа – 68 часов.

Общая трудоёмкость дисциплины составляет 180 часов, 5 зачетных единиц.

2.2. Основные разделы дисциплины:

1. Теоретическая информатика. Основы программирования.
2. Программное обеспечение ЭВМ. Интернет.

3. Планируемые результаты обучения (перечень компетенций)

Процесс изучения дисциплины направлен на формирование следующих компетенций:

ОПК-2 – способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

ОПК-3 – способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.

4. Вид промежуточной аттестации: зачет, 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.)

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО
КОНТРОЛЯ УСПЕВАЕМОСТИ ПО ДИСЦИПЛИНЕ**

1. Описание фонда оценочных средств (паспорт)

№ компетенции	Элемент компетенции	Раздел	Тема	ФОС
ОПК-2	- 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. Теоретическая информатика. Основы программирования. 2. Программное обеспечение ЭВМ. Интернет.	1.1. Информатика и информация. 1.2. Структура ЭВМ. Поколения ЭВМ. 1.3. Алгоритм 1.4. Язык программирования QBASIC 2.1. Операционные системы. 2.2. Текстовый редактор. 2.3. Табличный процессор. 2.4. Архиваторы. Антивирусы. Интернет.	Отчет по ЛР 1...11
ОПК-3	- способность использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.			

2. Описание показателей и критериев оценивания компетенций

Показатели	Оценка	Критерии
знать: <i>ОПК-2</i> - основы информационной и библиографической культуры, основные информационно-коммуникационные технологии и основные требования информационной безопасности; <i>ОПК-3</i> - современные информационные технологии, прикладные программные средства используемые при решении задач профессиональной деятельности; уметь: <i>ОПК-2</i> - решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учётом основных требований информационной безопасности; <i>ОПК-3</i> - использовать современные информационные технологии, прикладные программные средства используемые при решении задач профессиональной деятельности; Владеть: <i>ОПК-2</i> - способами применения информационно-коммуникационных технологий для решения стандартных профессиональных задач. <i>ОПК-3</i> - способностью использовать современные информационные технологии, прикладные программные средства при решении задач профессиональной деятельности.	зачтено	- даны исчерпывающие и обоснованные ответы на все поставленные вопросы; - ответы изложены грамотно, уверенно, логично, последовательно; - опираясь на усвоенные знания, четко увязывает научные положения с практической деятельностью; - свободно владеет основными понятиями дисциплины.
	не зачтено	- допускает существенные ошибки и неточности при ответе на поставленные вопросы; - испытывает трудности в практическом применении полученных знаний; - не может аргументировать научные положения; - не владеет системой основных понятий дисциплины.

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 15.03.05 Конструкторско-технологическое обеспечение машиностроительных производств от 11 августа 2016 г № 1000

для набора 2015 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «03» июля 2018г. № 413,

для набора 2016 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «06» октября 2016 г. № 684,

для набора 2017 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «06» марта 2017 г. № 125,

для набора 2018 года и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «12» марта 2018г. № 130.

Программу составил:

Архипов П.В., доцент кафедры МиТ, канд. техн. 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 МиТ _____ Е.А. Слепенко

Директор библиотеки _____ Т.Ф. Сотник

Рабочая программа одобрена методической комиссией факультета МФ

от «14» декабря 2018 г., протокол № 4

Председатель методической комиссии факультета МФ _____ Г.Н. Плеханов

СОГЛАСОВАНО:

Начальник учебно-методического управления _____ Г.П. Нежевец

Регистрационный № 434