

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Амурский государственный университет»

Кафедра Конструирования и технологии одежды

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ**

**Информатика**

Специальности 260704.65 – Технология текстильных изделий  
260901.65 – Технология швейных изделий  
260902.65 – Конструирование швейных изделий

Благовещенск 2012

УМКД разработан канд.техн.наук, доцентами кафедры конструирования и технологии одежды Абакумовой Ириной Валентиновной и Божук Галиной Анатольевной

Рассмотрен и рекомендован на заседании кафедры

Протокол заседания кафедры от «\_\_\_» \_\_\_\_\_ 2012 г. № \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_ / И.В. Абакумова /

**УТВЕРЖДЕН**

Протокол заседания УМСС по специальностям 260704.65 – Технология текстильных изделий, 260901.65 – Технология швейных изделий, 260902.65 – Конструирование швейных изделий

от «\_\_\_» \_\_\_\_\_ 2012 г. № \_\_\_\_\_

Председатель УМСС \_\_\_\_\_ /И.В.Абакумова/

# 1 РАБОЧАЯ ПРОГРАММА

## 1.1 цели и задачи освоения дисциплины

**Цель дисциплины:** сформировать у студентов комплекс теоретических знаний и практических навыков в области современных информационных технологий с учетом тенденций их развития.

**Задачи дисциплины** – научить студентов:

- принципам применения современных информационных технологий в профессиональной деятельности;
- практическим навыкам работы с современной компьютерной техникой;
- работать с пакетами прикладных программ общего назначения и специализированными профессионально-ориентированными программными средствами.

## 1.2 Место дисциплины в структуре ОПП

Дисциплина «Информатика» входит в базовую часть математического и естественнонаучного цикла основной образовательной программы высшего профессионального образования по специальностям 260704.65 – Технология текстильных изделий, 260901.65 – Технология швейных изделий, 260902.65 – Конструирование швейных изделий и реализуется на начальной стадии освоения данного цикла в 1,2 семестрах.

## 1.3 Требования к результатам освоения дисциплины

В результате изучения учебной дисциплины «» студент должен:

**знать:**

- принципы и методы обработки, хранения и передачи информации;
- основы алгоритмизации и программирования;
- общие сведения о пакетах прикладных программ,
- структуру локальных и глобальных компьютерных сетей.

**уметь:**

- использовать современные компьютерные информационные технологии обработки информации,
- обмениваться информацией в сетях,
- соблюдать основные требования информационной безопасности при решении профессиональных задач.

**владеть:**

- навыками в области информатики, применения специальных и прикладных программных средств для обработки текстовой и графической информации;
- основами автоматизации решения производственных и экономических задач;
- приемами антивирусной защиты, работы в компьютерных сетях.

## 1.4 Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины «Информатика» составляет \_220 часов

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				лекц.	лаб.раб.	сам.	
1	Данные – как форма документирования информационных процессов	1	1-4	4	2	6	индивидуальные задания
2	Современный компьютер и его технические и программные средства реализации информационных процессов	1	5-8	4	2	6	индивидуальные задания
3	Операционные системы	1	9-10	2	6	8	контр. работа реферат
4	Пакеты прикладных программ общего назначения	1	11-12	4	16	18	контр. работа зачет
5	Алгоритмы и алгоритмизация. Визуализация алгоритмов	1	13-18	4	10	16	контр. работа реферат зачет
		1	18	18	36	31	зачет
6	Обзор языков высокого уровня	2	1-2	2	4	6	индивидуальные задания
7	Технология программирования	2	3-8	6	20	26	контр. работа реферат
8	Принципы построения информационно-вычислительных сетей ЭВМ	2	9-12	4	4	8	контр. работа реферат
9	Основы и методы защиты информации в вычислительных системах и сетях	2	13-16	4	2	6	реферат, подготовка к экзамену
10	Специализированные профессионально-ориентированные программные средства	2	17-18	2	6	8	индивидуальные задания, реферат, подготовка к экзамену
		2	18	18	36	54	экзамен
Всего:				36	90	94	

## 1.5. Содержание разделов и тем дисциплины

### 1.5.1. Лекции

#### Тема 1. Данные - как форма документирования информационных процессов

Понятие информации. Сигналы и данные - как форма представления сведений, понятие информации. Кодирование данных и операции над данными. Единицы представления, измерения и хранения данных. Предмет и задачи информатики, ее место среди других наук. Новые информационные технологии как основа повышения эффективности управле-

ния производством и экономикой. Информационные технологии в области легкой промышленности. История, перспективы и темпы развития компьютерных технологий. Общая характеристика процессов сбора, передачи, обработки и накопления информации.

#### Тема 2. Современный компьютер и его технические и программные средства реализации информационных процессов

Системы счисления. Арифметические и логические основы выполнения операций над данными. Основные компоненты ЭВМ и принцип действия компьютера. Центральный процессор, оперативная память, системная магистраль, внешняя память и устройства ввода/вывода. Классификация ЭВМ. Типовая архитектура аппаратных средств IBM-совместимых персональных компьютеров (PC). Назначение и классификация программного обеспечения ЭВМ, иерархия программных средств. Типовая архитектура программных средств IBM-совместимых персональных компьютеров (PC). Назначение и виды операционных системы ЭВМ.

#### Тема 3. Операционные системы

Назначение и виды операционных системы ЭВМ. Основные функции ОС по обеспечению интерфейса пользователя, программно-аппаратного и программного интерфейса. Дополнительные функции операционных систем. Основные возможности ОС, структура ПО и требования к аппаратному обеспечению. Работа с файлами и каталогами. Программы-утилиты и программы-оболочки. Архивация файлов.

#### Тема 4. Пакеты прикладных программ общего назначения

Классификация и области применения прикладных программ общего назначения. Автоматизация задач делопроизводства. Стандартные средства пакета MS Office. Функциональные возможности и области применения текстовых редакторов и текстовых процессоров. Текстовые и графические редакторы.

#### Тема 5. Алгоритмы и алгоритмизация. Визуализация алгоритмов

Понятие алгоритма и алгоритмической системы. Две формы представления алгоритмов: визуальная и текстовая. Преобразование алгоритмов из визуальной формы в текстовую и обратно. Визуализация алгоритмов и блок-схем. Линейные, разветвленные и циклические алгоритмы. Вложенные и параллельные алгоритмы. Логические элементы и базовые управляющие структуры визуального структурного программирования.

#### Тема 6. Обзор языков высокого уровня

Компьютер как исполнитель алгоритмов. Программа как изображение алгоритма в терминах команд, управляющих работой компьютера. Коды, ассемблеры, языки высокого уровня. Понятие языка высокого уровня. Синтаксис и семантика. Элементы и структуры данных, алфавит, имена, выражения, операции, операторы, структура программ, аппарат подпрограмм, реализация логических структур языка программирования Паскаль.

#### Тема 7. Технология программирования

Моделирование как средство поддержки принятия решений в профессиональной области. Основные этапы разработки программ решения задачи на ЭВМ. Модели решения функциональных и вычислительных задач. Архитектура программных систем. Программное обеспечение и технологии программирования. Основы создания программного продукта. Компиляторы и интерпретаторы. Системы программирования и их компоненты. Основные методы и технологии программирования. Среда Паскаль. Типы данных. Операторы. Ввод-вывод данных. Массивы, циклы. Условные операторы. Встроенные функции.

#### Тема 8. Принципы построения информационно-вычислительных сетей ЭВМ

Локальные и глобальные сети ЭВМ. Понятие информационно-вычислительных сетей (ИВС) и сетей обмена данными (СОД), их основные категории и базовые технологии. Основные положения эталонной модели взаимодействия открытых систем. Типовая структура тракта передачи данных. Каналы связи, используемые для передачи данных. Коммутация в СОД. Характеристики ЛВС (локальные вычислительные сети), типы их топологий, методы доступа. Сетевое оборудование для ЛВС: кабели, адаптеры и повторители, мосты и маршрутизаторы. Работа в локальной сети Windows (Windows NT). Техноло-

гии городских и глобальных вычислительных сетей. Способы подключения к Internet. Услуги компьютерной сети Internet. Работа в глобальной сети Internet, использование электронной почты, методов доступа FTP, WWW и др. Работа с WWW браузерами.

#### Тема 9. Основы и методы защиты информации в вычислительных системах и сетях

Основы защиты информации и сведений, составляющих государственную тайну. Основы и методы защиты информации. Информационная безопасность (ИБ) и ее составляющие. Справочные информационные системы. Угрозы безопасности информации и их классификация. Основные виды защищаемой информации. Проблемы ИБ в мировом сообществе. Законодательные и иные правовые акты РФ, регулирующие правовые отношения в сфере ИБ и защиты государственной тайны. Административно-правовая и уголовная ответственность в информационной сфере. Защита информации в локальных компьютерных сетях, антивирусная защита. Специфика обработки конфиденциальной информации в компьютерных системах.

#### Тема 10. Специализированные профессионально - ориентированные программные средства

Пакеты программ для организации работ в офисе. Обзор специализированных программных средств: интегрированные системы делопроизводства, бухгалтерские системы, финансовые аналитические системы, геоинформационные системы, системы видеомонтажа, обучающие, развивающие, справочные развлекательные системы и программы. Математические пакеты. Системы автоматизированного проектирования (САД-системы), автоматизированные системы управления технологическими процессами (АСУТП). Понятие экспертной системы.

##### *1.5.2. Лабораторные работы.*

1. Функциональные блоки компьютера и их назначение. Овладение навыками работы с клавиатурой, мышью, экраном и принтером.

2. Ознакомление с аппаратной частью персонального компьютера

3. Работа с операционной системой. Операционная система MS Windows. Загрузка и ознакомление с графическим интерфейсом, возможности его настройки. Управление прикладными окнами. Варианты запуска приложений. Работа с учебником по MS Windows. Работа с программами «Мой компьютер» и «Проводник». Создание папок. Основные операции с файлами и папками.

4. Работа с текстовым редактором. Ввод, редактирование и форматирование текста. Работа с таблицами в MS Word. Создание таблиц. Основные приемы редактирования и форматирования таблиц. Вычисления в таблицах. Работа с графическими объектами в MS Word. Вставка рисунков. Фигурный текст. Рисование схем.

5. Проектирование визуального алгоритма на экране компьютера с помощью графического редактора и вывод его на плоттер формата А1 или принтер. Работа с встроенным графическим редактором Paint. Инструменты и команды. Рисование графических примитивов и кривых. Работа с буфером обмена.

6. Система программирования Паскаль. Среда Паскаль. Типы данных. Операторы Паскаля. Ввод программы.

7. Структура программы. Основные разделы программы. Элементы программирования. Разработка программ для заданных условий. Массивы, циклы. Условные операторы. Встроенные функции. Ввод-вывод данных..

8. Основы работы в Интернет. Работа с WWW браузерами.

9. Работа с антивирусными программами - проверка диска на наличие вируса.

10. Системы автоматизированного проектирования (САД-системы).

#### **1.6. Самостоятельная работа**

Самостоятельная работа студентов по дисциплине включает:

- самостоятельное изучение теоретических разделов дисциплины по заданию лектора;

- повторение и углубленное изучение лекционного материала;
- написание реферата;
- подготовку и выполнение лабораторных работ;
- выполнение индивидуальных заданий, контрольных работ и расчетов на ПК;
- подготовку к зачету и экзамену.

*Темы для самостоятельного изучения:*

- 1) Кодирование информации.
- 2) Двоичная система счисления.
- 3) История развития вычислительных средств.
- 4) Аппаратура компьютера.
- 5) Архитектура компьютера.
- 6) Системное и прикладное программное обеспечение.
- 7) Понятие об операционной системе.
- 8) Прикладное программное обеспечение.
- 9) Машинная графика. Назначение и основные функции.
- 10) Системы управления базами данных (СУБД). Назначение и основные функции.
- 11) Электронные таблицы. Назначение и основные функции.
- 12) Понятие программного продукта. Жизненный цикл программного обеспечения.
- 13) Задачи, решаемые с помощью баз данных.
- 14) Экономические, организационные и правовые вопросы создания программного и информационного обеспечения. Понятие интеллектуальной собственности.
- 15) Понятие телекоммуникации. Компьютерные сети как средство реализации практических потребностей.
- 16) Интегрированные автоматизированные системы
- 17) Организационные меры, инженерно-технические и иные методы защиты информации в том числе сведений, составляющих государственную тайну.
- 18) Угрозы безопасности информации и их классификация. Основные виды защищаемой информации. Проблемы ИБ в мировом сообществе.
- 19) Интеллектуальные системы решения вычислительных задач и моделирования.
- 20) Искусственный интеллект и интеллектуальные системы.
- 21) Тематика работ для самостоятельного изучения сообщается студентам на первой лекции. Материал может быть подготовлен в виде реферата и (или) представлен в виде доклада на соответствующей лекции или лабораторном занятии.

### **1.7 Курсовая работа**

Курсовая работа предусмотрена для закрепления знаний по основам информатики и навыков программирования в среде Турбо Паскаль.

Пояснительная записка к курсовой работе выполняется (набирается) на компьютере в текстовом редакторе WORD.

Курсовая работа содержит необходимые в соответствии с требованиями нормоконтроля разделы.

#### *Структура курсовой работы*

Введение (указываются цели и задачи курсовой работы)

1 Стандартный модуль Crt.

1.1. ...

1.2. ...

...

2 Стандартный модуль Graph.

2.1. ...

2.2. ...

...Программирование на языке Turbo Pascal

...

1.1. Создание программ для расчета значения функции с заданным диапазоном и шагом изменения аргумента

...

- 1.1.1. Программа с использованием оператора цикла с постусловием ...
  - 1.1.2. Программа с использованием оператора цикла с предусловием ...
  - 1.1.3. Программа с использованием процедур и функций модуля Crt
  - 1.2. Создание программ для расчета статистических характеристик
    - 1.2.1. Программа для расчета статистических характеристик с файловым вводом данных
    - 1.2.2. Программа для расчета статистических характеристик с файловым выводом данных
    - 1.2.3. Программа для расчета статистических характеристик с использованием процедур и функций модуля Graph
  - 1.3. Создание программы с использованием условных операторов
  - 1.4. Создание программы с использованием оператора цикла FOR
  - 1.5. Создание программы для реализации процесса тестирования ...
    - 1.5.1. Программа тестирования с вопросами, представленными в файле исходных данных
    - 1.5.2. Программа тестирования с вопросами, представленными в программе
    - 1.5.3. Программа тестирования с использованием процедур и функций модуля Crt
- Заключение (делаются выводы по работе)  
 Список использованных источников  
 Приложения (приводятся распечатки всех программ)

### 1.8 Образовательные технологии

Вид занятий	Образовательные технологии, средства и методы
Лекционные	Неимитационные активные инновационные методы: коллективные формы работы в виде диалога и полилога, методы коллективного взаимообучения, коллективного способа обучения. проблемные лекции, лекция-визуализация, лекция-консультация. Информационные технологии: мультимедийное обучение, Информационные системы: электронные библиотеки, электронные базы учебно-методических ресурсов
Лабораторные	Неигровые имитационные методы: индивидуальные методы обучения с выполнением индивидуальных заданий (алгоритмизированных, творческих, поисковых), при специальном обучении поисковым процедурам, при подготовке рефератов, докладов, сообщений, методы группового решения творческих задач, метод развивающейся кооперации, групповые формы обучения с использованием методов обучения в команде, работы в парах сменного состава. Информационные технологии: мультимедийное обучение, интерактивное обучение (компьютерные программы); сетевые компьютерные технологии (Интернет) Информационные системы: электронные библиотеки, электронные базы учебно-методических ресурсов
Самостоятельная	Информационные технологии: мультимедийное обучение, интерактивное обучение (компьютерные программы); сетевые компьютерные технологии (Интернет) Информационные системы: электронные библиотеки, электронные базы учебно-методических ресурсов



В соответствии с требованиями ГОС ВПО в учебном процессе предусмотрено использование активных форм проведения занятий, развивающих коммуникативные способности и речь обучающихся, направленных на их привлечение к самостоятельной познавательной деятельности, вызывающих личностный интерес к решению проблемных задач.

### **1.9. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов**

#### *1.9.1. Вопросы к зачету:*

- 1) Основные компоненты ПЭВМ.
- 2) Структура программного обеспечения ЭВМ.
- 3) Логические основы ЭВМ.
- 4) Видеосистема. ПЭВМ.
- 5) Накопители на жестких и гибких магнитных дисках.
- 6) Принтеры. Типы. Способы печати.
- 7) Работа в сети.
- 8) Системы счисления.
- 9) Перевод чисел из одной системы счисления в другую.
- 10) Функции операционной системы.
- 11) Текстовый редактор. Назначение и основные функции.
- 12) Локальные и телекоммуникационные компьютерные сети. Компьютерная почта.
- 13) Информатизация общества. Развитие вычислительной техники.
- 14) Информация. Единицы измерения информации.
- 15) Понятия алгоритма. Исполнители алгоритмов.
- 16) Линейные алгоритмы.
- 17) Разветвляющийся алгоритм.
- 18) Циклический алгоритм.
- 19) Конструирование алгоритмов методом последовательной детализации. Вспомогательный алгоритм.
- 20) Этапы решения задач на ЭВМ.
- 21) Антивирусные программы и программы-архиваторы.
- 22) *1.9.2. Вопросы к экзамену:*
- 23) Понятие информации. Свойства информации. Место и роль понятия «информация» в курсе информатики.
- 24) Кодированная информация.
- 25) Понятие носителя информации. Формы представления и передачи информации.
- 26) Назначение компьютера.
- 27) История развития вычислительных средств.
- 28) Основные функциональные части компьютера.
- 29) Основные принципы работы компьютера.
- 30) Программа как последовательность действий компьютера. Понятие о машинном языке и языке Ассемблер.
- 31) Аппаратура компьютера.
- 32) Архитектура компьютера.
- 33) Центральные устройства компьютера.
- 34) Понятие алгоритма и алгоритмической системы.
- 35) Линейные, разветвленные и циклические алгоритмы. Вложенные и параллельные алгоритмы.
- 36) Компьютер как исполнитель алгоритмов.
- 37) Программа как изображение алгоритма в терминах команд, управляющих работой компьютера.
- 38) Коды, ассемблеры, языки высокого уровня. Трансляция и компоновка.

- 39) Программное обеспечение компьютера.
- 40) Системное и прикладное программное обеспечение.
- 41) Понятие об операционной системе.
- 42) Прикладное программное обеспечение.
- 43) Понятие языка высокого уровня. Синтаксис и семантика. Элементы и структуры данных, алфавит, имена, выражения, операции, операторы языка программирования Паскаль
- 44) Структура программ, аппарат подпрограмм, реализация логических структур языка программирования Паскаль.
- 45) Понятие программного продукта. Жизненный цикл программного обеспечения.
- 46) Задачи, решаемые с помощью баз данных.
- 47) Понятие телекоммуникации. Компьютерные сети как средство реализации практических потребностей.
- 48) Локальные сети и глобальные сети: принципы построения, архитектура, основные компоненты, их назначение и функции.
- 49) Всемирная компьютерная сеть ИНТЕРНЕТ. Ее возможности.
- 50) Модели решения функциональных и вычислительных задач.
- 51) Компьютерная графика
- 52) Системы геометрического моделирования.
- 53) Интегрированные автоматизированные системы
- 54) Угрозы безопасности информации и их классификация. Основные виды защищаемой информации. Проблемы ИБ в мировом сообществе.
- 55) Организационные меры, инженерно-технические и иные методы защиты информации в том числе сведений, составляющих государственную тайну.
- 56) Защита информации в локальных компьютерных сетях, антивирусная защита.

### 1.9.3. Пример варианта контрольной работы

#### Задание 1

Работа выполняется в текстовом редакторе MS Word:

1. Составьте краткий реферат (5-10стр.) на тему:

*История развития информатики.*

Оформите реферат, используя 2-3 своих стиля.

Расставьте номера страниц в правом верхнем углу.

Задайте колонтитул:

- для первой страницы – тема реферата;
- для четной страницы верхний колонтитул – реферат подготовил ФИО студента, № группы, факультет;
- нижний колонтитул – документ создан – текущая дата;
- для нечетной страницы верхний колонтитул – контрольная работа №1; нижний колонтитул – документ создан – имя файла.

2. Разработайте в виде таблицы годовой бизнес-план выполнения ремонта офиса, в таблице отразите вид деятельности, исполнителей, финансирование по месяцам, затраты по кварталам и за год. Выполните соответствующие расчеты.

3. Создайте следующую формулу:

$$y = \begin{cases} 5\alpha^3 + \frac{4x + tg(x^2 + \beta)^2}{25} & x < 1 \\ (\alpha x + b)^4 + 10\sqrt[3]{|\sin x^2|} & \text{ïðå } 1 \leq \delta \leq 3 \\ \ln(b^3 + \sqrt{x+a}) & x > 3 \end{cases}$$

4. Используя WordArt, графику, рисунки, символы, списки подготовьте:

- текст объявления на произвольную тему;
- визитную карточку.
- 5. Составьте визуальные алгоритмы для следующих задач:
  - найти периметр и площадь прямоугольника (ширину и длину вводить с клавиатуры);
  - для двух чисел X, Y определить, являются ли они корнями уравнения:
 
$$A \cdot P^2 + B \cdot P + C = 0;$$
  - вычислить число в факториале  $Y = X!$ .

*Примечание:* Контрольная работа должна быть распечатана и представлена в электронном и печатном виде.

### Задание 2

Уравнение прямой:

$$Y = ax + b$$

Для определения коэффициентов a и b, рассмотрим точки пересечения прямой с осями X и Y.

1. Точка B (0;2)

Подставим координаты т.В в уравнение прямой и определим значение коэффициента b:

$$x=0, y=2$$

$$2 = a \cdot 0 + b$$

$$b = 2$$

2. Точка A (2;0)

Подставим координаты т.А и значение коэффициента  $b=2$  в уравнение прямой и определим значение коэффициента a:

$$0 = a \cdot 2 + 2$$

$$a = -1$$

Следовательно, уравнение прямой будет иметь следующий вид:

$$y = -x + 2.$$

Для всех точек, находящихся снизу от прямой, будет выполняться следующее неравенство, определяющее условие заштрихованной области:

$$y \leq -x + 2.$$

### 1.9.4. Курсовая работа

Курсовая работа предусмотрена для закрепления знаний по основам информатики и навыков программирования в среде Турбо Паскаль.

Пояснительная записка к курсовой работе выполняется (набирается) на компьютере в текстовом редакторе WORD.

Курсовая работа содержит необходимые в соответствии с требованиями нормоконтроля разделы.

### Структура курсовой работы

Введение (указываются цели и задачи курсовой работы)

1 Стандартный модуль Crt.

2 Стандартный модуль Craph.

3 Программирование на языке Turbo Pascal

...

3.1 Создание программ для расчета значения функции с заданным диапазоном и шагом изменения аргумента

3.2 Программа с использованием оператора цикла с постусловием

3.3 Программа с использованием оператора цикла с предусловием

3.4 Программа с использованием процедур и функций модуля Crt

4 Создание программ для расчета статистических характеристик

4.1 Программа для расчета статистических характеристик с файловым вводом данных

- 4.2 Программа для расчета статистических характеристик с файловым выводом данных
- 4.3 Программа для расчета статистических характеристик с использованием процедур и функций модуля Graph
- 4.4 Создание программы с использованием условных операторов
- 4.5 Создание программы с использованием оператора цикла FOR
- 5 Создание программы для реализации процесса тестирования ...
  - 5.1 Программа тестирования с вопросами, представленными в файле исходных данных
  - 5.2 Программа тестирования с вопросами, представленными в программе
  - 5.3 Программа тестирования с использованием процедур и функций модуля Crt

Заключение (делаются выводы по работе)

Список использованных источников

Приложения (приводятся распечатки всех программ)

### 1.10. Учебно-методическое и информационное обеспечение дисциплины

#### 1.10.1 основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек.Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

#### 1.10.2 дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.
- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапорев С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапорев.- СПб.: БХВ-Петербург, 2008.- 469с.
- 4) Информатика и программирование: компьютерный практикум: учеб.пособие: рек УМО/ А.Н.Гуда и др.; под общ.ред. В.И.Колесникова. – М.: Дашков и К, 2009, 2010.- 234с.
- 5) Ярочкин В.И. Информационная безопасность: учеб.: рек. Мин.обр.РФ/ М.: Академический Проект, 2008. – 544с.
- 6) Абакумова И.В., Тибенко Т.А., Сухова Т.Н. Обработка данных средствами Excel. Учебно-методическое пособие. Амурский гос.ун-т, Благовещенск, 2006.

#### 1.10.3 Периодические издания:

Журналы: «Информатика», «Информационно-управляющие системы», «Информационные системы и технологии», «Информационные технологии и вычислительные системы»

#### 1.10.4 Программное обеспечение и Интернет-ресурсы:

№	Наименование ресурса	Краткая характеристика
1	<a href="http://www.iqlib.ru">http://www.iqlib.ru</a>	Интернет библиотека образовательных изданий, в которой собраны электронные учебники, справочные и учебные пособия. Удобный поиск по ключевым словам, отдельным темам и отраслям знаний
2	Самоучитель по программированию на языках Free Pascal и Lazarus <a href="http://teacher.ucoz.net/Lecture/Pascal/Lazarus.pdf/">http://teacher.ucoz.net/Lecture/Pascal/Lazarus.pdf/</a>	Содержит теоретический материал по программированию на языках Free Pascal и Lazarus
3	Практикум по информатике	Содержит практические задания по

	<a href="http://www.login.ru/books/13735">http://www.login.ru/books/13735</a>	информатике и варианты тестов
4	Информатика. Практикум по технологии работы на компьютере. Под ред. Макаровой Н.В. <a href="http://www.alleng.ru/d/comp/comp-40.htm">http://www.alleng.ru/d/comp/comp-40.htm</a>	Содержит различные технологии работы на компьютере, комплекс практических заданий

### 1.11 Материально-техническое обеспечение дисциплины

При освоении учебной дисциплины рекомендуется использование аудитории с мультимедийным обеспечением для проведения лекционных занятий, для демонстрации домашнего задания, выполняющего студентами самостоятельно. Для проведения лабораторных занятий – компьютерный класс.

## 2 КРАТКОЕ ИЗЛОЖЕНИЕ ПРОГРАММНОГО МАТЕРИАЛА

### Тема 1. Информация (2 часа)

*Цель:* дать понятия об информации

*План:*

- 1) Понятие об информации.
- 2) Язык как способ представления информации.
- 3) Виды и свойства информации.
- 4) Информационные процессы и технологии.
- 5) Формы представления информации.
- 6) Понятие количества информации.
- 7) Кодирование информации.
- 8) Носители информации

*Ключевые вопросы:*

Информация — сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы (живые организмы, управляющие машины и др.) в процессе жизнедеятельности и работы.

Возможность автоматизированной обработки информации основывается на том, что обработка информации не подразумевает ее осмысления.

Хранение информации — это способ распространения информации в пространстве и времени.

Основные хранилища информации:

- 1) для человека: память.
- 2) для общества: библиотеки, видеотеки, фонотеки, архивы, патентные бюро, музеи, картинные галереи.
- 3) компьютерные хранилища: базы и банки данных, информационно - поисковые системы, электронные энциклопедии, медиатеки.

Информационная система — это хранилище информации, снабженное процедурами ввода, поиска и размещения и выдачи информации. Наличие таких процедур — главная особенность информационных систем, отличающих их от простых скоплений информационных материалов.

*Литература:*

Основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек. Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

Дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.

- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапорев С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапорев.- СПб.: БХВ-Петербург, 2008.- 469с.
- 4) Информатика и программирование: компьютерный практикум: учеб.пособие: рек УМО/ А.Н.Гуда и др.; под общ.ред. В.И.Колесникова. – М.: Дашков и К, 2009, 2010.- 234с.

*Выводы по теме:*

Информация — это отражение внешнего мира, а он существует независимо от нашего сознания и желания. Поэтому в качестве свойства информации можно выделить ее объективность. Информация объективна, если она не зависит от чьего-либо мнения, суждения.

Классификация информации:

1) По способам восприятия: визуальная; аудиальная; тактильная; обонятельная, вкусовая.

2) По форме представления: текстовая; числовая; графическая; музыкальная; комбинированная.

3) По общественному значению: - Массовая: обыденная; общественно-политическая; эстетическая. Специальная: научная; производственная; техническая; управленческая. Личная: знания; умения; навыки; интуиция.

Информация нужна для того, чтобы принимать правильные решения. Качественные признаки информации.

Информация достоверна, если она отражает истинное положение дел. Объективная информация всегда достоверна, но достоверная информация может быть как объективной, так и субъективной. Достоверная информация помогает принять нам правильное решение. Недостоверной информация может быть по следующим причинам: преднамеренное искажение (дезинформация); искажение в результате воздействия помех («испорченный телефон»); когда значение реального факта преуменьшается или преувеличивается (слухи, рыбацкие истории).

Актуальность — важность, существенность для настоящего времени. Только вовремя полученная информация может принести необходимую пользу. Неактуальной информация может быть по двум причинам: она может быть устаревшей.

Информационные процессы - процессы, связанные с поиском, обработкой, хранением и передачей информации.

Поиск информации — это извлечение хранимой информации. Существуют ручной и автоматизированный метод поиска информации в хранилищах.

Методы поиска информации: непосредственное наблюдение; общение со специалистами по интересующему вас вопросу; чтение соответствующей литературы; просмотр теле-, видеопрограмм; прослушивание радиопередач и аудиокассет; работа в библиотеках, архивах; запрос к информационным системам, базам и банкам компьютерных данных; другие методы.

## Тема 2 - Принцип работы компьютера (3 часа)

*Цель:* разъяснить принцип работы компьютера

*План:*

- 1) Устройство компьютера
- 2) Функции памяти
- 3) Функции процессора

*Выводы по теме:*

Компьютер (англ. computer — вычислитель) представляет собой программируемое электронное устройство, способное обрабатывать данные и производить вычисления, а также выполнять другие задачи манипулирования символами

Любая компьютерная программа представляет собой последовательность отдельных команд.

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

Структуры компьютера основаны на общих логических принципах, позволяющих выделить в любом компьютере следующие главные устройства:

- память (запоминающее устройство, ЗУ), состоящую из перенумерованных ячеек;
- процессор, включающий в себя устройство управления (УУ) и арифметико-логическое устройство (АЛУ);
- устройство ввода;
- устройство вывода.
- Функции памяти:
- приём информации из других устройств;
- запоминание информации;
- выдача информации по запросу в другие устройства машины.

Функции процессора:

- обработка данных по заданной программе путем выполнения арифметических и логических операций;

- программное управление работой устройств компьютера.

Регистр выполняет функцию кратковременного хранения числа или команды. Над содержимым некоторых регистров специальные электронные схемы могут выполнять некоторые манипуляции. Например, "вырезать" отдельные части команды для последующего их использования или выполнять определенные арифметические операции над числами.

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

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

Интерфейс — это средство сопряжения двух устройств, в котором все физические и логические параметры согласуются между собой.

Порты устройств представляют собой некие электронные схемы, содержащие один или несколько регистров ввода-вывода и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора.

Современный персональный компьютер состоит из нескольких основных конструктивных компонент:

- системного блока;
- монитора;
- клавиатуры;
- манипуляторов.

В системном блоке размещаются:

- блок питания;
- накопитель на жёстких магнитных дисках;
- накопитель на гибких магнитных дисках;
- системная плата;
- платы расширения;
- накопитель CD-ROM;
- и др.

Устройство системного блока:

1 — Системная плата.

2 — Разъём дополнительного второго процессора.

3 — Центральный процессор с радиатором для отвода тепла.

- 4 — Разъёмы оперативной памяти.
- 5 — Накопитель на гибких магнитных дисках.
- 6 — Накопитель CD-ROM.
- 7 — Сетевая карта.
- 8 — Графический акселератор.
- 9 — Блок питания, преобразующий переменное напряжение электросети в постоянное напряжение различной полярности и величины, необходимое для питания системной платы и внутренних устройств. Блок питания содержит вентилятор, создающий циркулирующие потоки воздуха для охлаждения системного блока.

*Литература:*

Основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек. Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

Дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.
- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапорев С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапорев.- СПб.: БХВ-Петербург, 2008.- 469с.
- 4) Информатика и программирование: компьютерный практикум: учеб.пособие: рек УМО/ А.Н.Гуда и др.; под общ.ред. В.И.Колесникова. – М.: Дашков и К, 2009, 2010.- 234с.

Тема 3. Аппаратура компьютера. Технические средства реализации информационных процессов. (2 часа)

*Цель занятия:* изучить основные технические средства реализации информационных процессов.

*Ключевые вопросы:*

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

*Входное устройство* служит для ввода в машину всей информации для решения задач. Эта информация состоит из некоторой программы и массива данных, с которыми программа будет работать. В подавляющем большинстве машин и программа, и данные закодированы как числа в бинарной системе счисления. Вся вводимая информация попадает в *запоминающее устройство* или память ЭВМ, где она хранится до момента, когда понадобится.

*Выходное устройство* выводит полученные результаты пользователю. Как правило, полученные данные сообщаются пользователю в удобной для него форме (например, в виде текста на языке, близком к обычному человеческому языку, графиков, рисунков и т. п.).

*Системный блок* представляет собой основной узел, внутри которого установлены наиболее важные компоненты. Устройства, находящиеся внутри системного блока, называют *внутренними*, а устройства, подключаемые к нему снаружи, называют *внешними*. Внешние дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных, также называют периферийными.



*Материнская плата (mother board)* – основная плата персонального компьютера, представляющая из себя лист стеклотекстолита, покрытый медной фольгой. Путем травления фольги получают тонкие медные проводники соединяющие электронные компоненты. На материнской плате размещаются:

- *процессор* – основная микросхема, выполняющая большинство математических и логических операций;
- *шины* – наборы проводников, по которым происходит обмен сигналами между внутренними устройствами компьютера;
- *оперативная память* (оперативное запоминающее устройство, ОЗУ) – набор микросхем, предназначенных для временного хранения данных, когда компьютер включен;
- *ПЗУ* (постоянное запоминающее устройство) – микросхема, предназначенная для длительного хранения данных, в том числе и когда компьютер выключен;
- *микропроцессорный комплект (чипсет)* – набор микросхем, управляющих работой внутренних устройств компьютера и определяющих основные функциональные возможности материнской платы;
- *разъемы* для подключения дополнительных устройств (слоты).

*Процессор* (микропроцессор, центральный процессор, CPU) – основная микросхема компьютера, в которой и производятся все вычисления. Он представляет из себя большую микросхему (например, размеры микропроцессора Pentium примерно 5\*5\*0,5 см), которую можно легко найти на материнской плате. На процессоре установлен большой медный ребристый радиатор, охлаждаемый вентилятором. Конструктивно процессор состоит из ячеек, в которых данные могут не только храниться, но и изменяться. Внутренние ячейки процессора называют регистрами. Важно также отметить, что данные, попавшие в некоторые регистры, рассматриваются не как данные, а как команды, управляющие обработкой данных в других регистрах. Среди регистров процессора есть и такие, которые в зависимости от своего содержания способны модифицировать исполнение команд. Таким образом, управляя засылкой данных в разные регистры процессора, можно управлять обработкой данных. На этом и основано исполнение программ.

*Оперативная память (RAM – Random Access Memory)* – это массив кристаллических ячеек, способных хранить данные. Существует много различных типов оперативной памяти, но с точки зрения физического принципа действия различают *динамическую память (DRAM)* и *статическую память (SRAM)*.

*Монитор* – устройство визуального представления данных. Это не единственно возможное, но главное устройство вывода. Его основными *потребительскими параметрами* являются: *размер* и *шаг маски экрана*, *максимальная частота регенерации изображения*, *класс защиты*.

*Клавиатура* – клавишное устройство управления персональным компьютером. Служит для ввода алфавитно-цифровых (знаковых) данных, а также команд управления. Комбинация монитора и клавиатуры обеспечивает простейший интерфейс пользователя. С помощью клавиатуры управляют компьютерной системой, а с помощью монитора получают от нее отклик.

*Мышь* – устройство управления манипуляторного типа. Представляет собой плоскую коробочку с двумя-тремя кнопками. Перемещение мыши по плоской поверхности синхронизировано с перемещением графического объекта (указателя мыши) на экране монитора.

*Сканеры* – устройства для оцифровки и ввода в компьютер изображений с бумажных копий - это старейших вид компьютерной периферии. Современные сканеры позволяют оцифровывать изображения даже объемных предметов и диапозитивов (слайдов).

*Литература:*

Основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек. Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

Дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.
- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапорев С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапорев.- СПб.: БХВ-Петербург, 2008.- 469с.
- 4) Информатика и программирование: компьютерный практикум: учеб.пособие: рек УМО/ А.Н.Гуда и др.; под общ.ред. В.И.Колесникова. – М.: Дашков и К, 2009, 2010.- 234с.

*Выводы по теме:*

*Структура компьютера* – это совокупность его функциональных элементов и связей между ними. Структура компьютера графически представляется в виде структурных схем, с помощью которых можно дать описание компьютера на любом уровне детализации.

*Входное устройство* служит для ввода в машину всей информации для решения задач. Эта информация состоит из некоторой программы и массива данных, с которыми программа будет работать.

*Монитор* – устройство визуального представления данных. Его основными *потребительскими параметрами* являются: *размер* и *шаг маски экрана*, *максимальная частота регенерации изображения*, *класс защиты*.

*Клавиатура* – клавишное устройство управления персональным компьютером. Служит для ввода алфавитно-цифровых (знаковых) данных, а также команд управления.

*Мышь* – устройство управления манипуляторного типа. Представляет собой плоскую коробочку с двумя-тремя кнопками.

*Сканеры* – устройства для оцифровки и ввода в компьютер изображений с бумажных копий - это старейших вид компьютерной периферии. Современные сканеры позволяют оцифровывать изображения даже объемных предметов и диапозитив

#### Тема 4. Алгоритмы и алгоритмизация. Визуализация алгоритмов (4 часа)

*Цель занятия:* дать понятие алгоритмов, разобрать визуализацию алгоритмов

*План:*

- 1) Анализ постановки задачи и ее предметной области
- 2) Формальное моделирование решения задачи
- 3) Практическое решение
- 4) Свойства алгоритма:
- 5) Метод структурной алгоритмизации.

*Ключевые вопросы:*

Свойства алгоритма:

- определенность – выполнив очередное действие, исполнитель должен точно знать, что ему делать дальше;

- дискретность – прежде, чем выполнить определенное действие, надо выполнить предыдущее;

- массовость – по одному и тому же алгоритму решаются однотипные задачи и неоднократно;

- понятность – алгоритм строится для конкретного исполнителя человеком и должен быть ему понятен. Это облегчает его проверку и модификацию при необходимости;

- результативность – алгоритм всегда должен приводить к результату.

#### *Литература:*

##### Основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек. Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

##### Дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.
- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапорев С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапорев.- СПб.: БХВ-Петербург, 2008.- 469с.
- 4) Информатика и программирование: компьютерный практикум: учеб.пособие: рек УМО/ А.Н.Гуда и др.; под общ.ред. В.И.Колесникова. – М.: Дашков и К, 2009, 2010.- 234с.

#### *Выводы по теме:*

Решение любой задачи является творческим процессом, который состоит из нескольких *последовательных этапов*. К ним относятся :

##### *А. Анализ постановки задачи и ее предметной области*

- понимание постановки и требований исходной задачи, определение предметной области, для которой поставлена задача,
- анализ предметной области, выявление данных, которые фиксируют входную и выходную информацию (определение их структуры и свойств ),  
определение отношений между данными, условий и ограничений, накладываемых на эти отношения,

##### *Б. Формальное моделирование решения задачи*

- выбор и применение формальной системы для описания модели предметной области и решения задачи,
- формирование основной идеи, выбор методов решения задачи,
- определение технологий, средств и исполнителя решения задачи, построение алгоритмов, реализующих выбранные методы,

##### *В. Практическое решение*

- применение выбранных методов и средств для решения ,
- анализ полученных результатов.

Эти этапы ориентированы для получения решения не отдельно взятой, конкретной задачи, а некоторого класса задач данного типа. Этап построения алгоритмов, реализующих выбранные методы решения задачи, детализирует и визуализирует процесс ее решения. Алгоритмизация позволяет уже на этом этапе оценить эффективность решения, уточнить методы решения для различных потоков входных данных и выявить некоторые ошибки.

На первом этапе уточняется постановка задачи, после чего выявляются отдельные явления, объекты, процессы, их связи и зависимости предметной области.

Здесь определяются такие понятия как *исходные* и *результатирующие данные*, абстрактно представляющие информацию о процессах предметной области реального мира, а также поясняются, каким образом из исходных данных могут быть получены результирующие.

*Исходные данные* должны быть полными, т.е. содержать данные, необходимые и достаточные для решения задачи. Если данные неполные, необходимо приложить допол-

нительные усилия для сбора дополнительных сведений; эта ситуация может также возникнуть на последующих этапах при выборе метода решения.

Различают исходные данные *трех видов*: постоянные, условно-постоянные и переменные.

*Постоянные исходные данные* - это данные, которые сохраняют свои значения в процессе решения задачи (математические константы, координаты неподвижных объектов) и не зависят от внешних факторов.

*Условно-постоянные данные* - это данные, которые могут иногда изменять свои значения; причем эти изменения не зависят от процесса решения задачи, а определяются внешними факторами (величина подоходного налога, курс валют, количество дней в году).

*Переменные данные* - это данные, которые изменяют свои значения в процессе решения задачи.

*Структурированные данные* отличаются от простых тем, что к ним применимо другое отношение: одно имя - много значений. Если все элементы, входящие в такую структуру, однотипны, то такая структура называется однородной, в противном случае - неоднородной. Классическим примером однородной структуры является *массив*, являющийся последовательностью однотипных значений, таких как, например, (2,51,3,7,88). Неоднородная структура в отличие от однородной содержит значения различных типов, относящихся к одному понятию или объекту, и, значит, такое структурированное данное несет в себе больше информации. Для представления неоднородных структур используют *запись*. *Запись* - это структура, предназначенная для представления данных различного типа. Запись состоит из поименованных полей, каждое из которых должно содержать значение определенного типа.

*Модель* - это упрощенное представление о реальном объекте, процессе или явлении.

*Моделирование* - построение моделей для исследования и изучения моделируемого объекта, процесса, явления с целью получения новой информации при решении конкретных задач.

*Алгоритм* - это точно определенная последовательность действий для некоторого исполнителя, выполняемых по строго определенным правилам и приводящих через некоторое количество шагов к решению задачи.

*Исполнитель алгоритмов* определяет элементарные действия, из которых формируется алгоритм. Отдельные действия, составляющие алгоритм, называются операциями. При этом под операцией понимается как какое-то единичное действие, например, сложение, так и группа взаимосвязанных действий.

Алгоритм моделирует решение задачи в виде точно определенной последовательности действий для некоторого исполнителя по преобразованию исходных данных в результирующие. Процесс составления алгоритмов называют *алгоритмизацией*.

Алгоритм, реализующий решение задачи, можно представить различными способами: с помощью *графического* или *текстового* описания, в виде *таблицы значений*.

*Графический способ* представления алгоритмов имеет ряд преимуществ благодаря визуальности и явному отображению процесса решения задачи. Алгоритмы, представленные графическими средствами, получили название *визуальные алгоритмы*.

*Текстовое описание* алгоритма является достаточно компактным и может быть реализовано на абстрактном или реальном языке программирования в виде программы для ЭВМ.

*Таблицы значений* представляют алгоритм неявно, как некоторое преобразование конкретных исходных данных в выходные. Табличный способ описания алгоритмов может быть с успехом применен для проверки правильности функционирования разработанного алгоритма на конкретных тестовых наборах входных данных, которые вместе с результатами выполнения алгоритма фиксируются в "таблицах трассировки".

Одним из системных методов разработки алгоритмов является *метод структурной алгоритмизации*. Этот метод основан на визуальном представлении алгоритма в виде последовательности управляющих структурных фрагментов. Выделяют *три базовые* управляющие процессом обработки информации *структуры: композицию, альтернативу и итерацию*. С помощью этих структур можно описать любые процессы обработки информации.

#### Тема 5. Программирование (2 часа)

*Цель:* дать понятия о программировании

*План:*

- 1) Что такое алгоритмы
- 2) Языки программирования

*Литература:*

Основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек. Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

Дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.
- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапорев С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапорев.- СПб.: БХВ-Петербург, 2008.- 469с.
- 4) Информатика и программирование: компьютерный практикум: учеб.пособие: рек УМО/ А.Н.Гуда и др.; под общ.ред. В.И.Колесникова. – М.: Дашков и К, 2009, 2010.- 234с.

*Выводы по теме:*

Под *алгоритмом* понимают постоянное и точное предписание (указание) исполнителю совершить определенную последовательность действий, направленных на достижение указанной цели или решение поставленной задачи.

*Языки программирования высокого уровня* существенно отличаются от машинно-ориентированных (низкого уровня) языков. Во-первых, машинная программа, в конечном счете, записывается с помощью лишь двух символов 0 и 1. Во-вторых, каждая ЭВМ имеет ограниченный набор машинных операций, ориентированных на структуру процессора. Как правило, этот набор состоит из сравнительно небольшого числа простейших операций, типа: переслать число в ячейку; считать число их ячейки; увеличить содержимое ячейки на +1 и т.п. Команда на машинном языке содержит очень ограниченный объем информации, поэтому она обычно определяет простейший обмен содержимого ячеек памяти, элементарные арифметические и логические операции. Команда содержит код и адреса ячеек, с содержимым которых выполняется закодированное действие.

Языки программирования, имитирующие естественные языки, обладающие укрупненными командами, ориентированными на решение прикладных содержательных задач, называются *языками высокого уровня*.

Языки программирования высокого уровня имеют следующие *достоинства*:

- алфавит языка значительно шире машинного, что делает его гораздо более выразительным и существенно повышает наглядность и понятность текста;

- набор операций, допустимых для использования, не зависит от набора машинных операций, а выбирается из соображений удобства формулирования алгоритмов решения задач определенного класса;
- конструкции команд (операторов) отражают содержательные виды обработки данных и задаются в удобном для человека виде;
- используется аппарат переменных и действий с ними;
- поддерживается широкий набор типов данных.

#### *Fortran*

Это первый компилируемый язык созданный Джимом Бэкусом в 50-е годы. Программисты, разрабатывавшие программы исключительно на ассемблере, выражали серьезное сомнение в возможности появления высокопроизводительного языка высокого уровня, поэтому основным критерием при разработке компиляторов Фортрана являлась эффективность исполняемого хода. Хотя в Фортране был впервые реализован ряд важнейших понятий программирования, удобство создания программ было принесено в жертву возможности получения эффективного машинного кода. Однако для этого языка было создано огромное количество библиотек ,начиная от статических комплексов и кончая пакетами управления спутниками, поэтому Фортран продолжает активно использоваться во многих организациях, а сейчас ведутся работы над очередным стандартом Фортрана F2k, который появился в 2000 году.

#### *Cobol*

Это компилируемый язык для применения в экономической области и решения бизнес задач, разработанный в начале 60-х годов. Он отличается большой "многословностью" -его операторы выглядят как обычные английские фразы. В Коболе были реализованы очень мощные средства работы с большими объемами данных ,хранящимися на различных внешних носителях. На этом языке создано очень много различных приложений, которые активно эксплуатируются и сегодня. *Algol*

#### *Pascal*

Язык Паскаль, созданный в конце 70-х годов основоположником множества идей современного программирования Никлаусом Виртом, во многом напоминает Алгол, но в нем ужесточен ряд требований к структуре программы и имеются возможности, позволяющие успешно применять его при создании крупных проектов.

#### *Basic*

Для этого языка имеются и компиляторы и интерпретаторы, а по популярности он занимает первое место в мире. Он создавался в конце 60-х годов в качестве учебного пособия и очень прост в изучении.

С помощью языка программирования создаётся не готовая программа, а только её текст, описывающий ранее разработанный алгоритм. Чтобы получить *работающую программу*, надо этот текст либо автоматически перевести в машинный код (для этого служат программы *компиляторы*) и затем использовать отдельно от исходного текста, либо сразу выполнять команды языка, указанные в тексте программы (этим занимаются программы-*интерпретаторы*).

*Интерпретатор* берёт очередной оператор языка из текста программы, анализирует его структуру и затем сразу исполняет (обычно после анализа оператор транслируется в некоторое промежуточное представление или даже машинный код для более эффективно дальнейшего исполнения). Только после того как текущий оператор успешно выполнен, интерпретатор перейдёт к следующему. При этом если один и тот же оператор будет выполняться в программе многократно, интерпретатор будет выполнять его так как, как будто встретил впервые. Вследствие этого программы, в которых требуется осуществить большой объём вычислений, будут выполняться медленно. Кроме того, для выполнения программы на другом компьютере там тоже должен стоять интерпретатор – ведь без него текст является просто набором символов.

*Компиляторы* полностью обрабатывают весь текст программы (он иногда называется исходный код) Они просматривают его в поиске синтаксических ошибок (иногда несколько раз), производят определенный смысловой анализ, а затем автоматически переводят (транслируют) на машинный язык - генерируют машинный код.

#### Тема 6. Программное обеспечение (3 часа)

*Цель:* рассмотреть программное обеспечение

*План:*

- 1) Программное обеспечение компьютера
- 2) Принцип организации файловой системы
- 3) Создание и именование файлов каталогов
- 4) Навигация по файловой структуре

*Литература:*

Основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек.Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

Дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.
- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапоров С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапоров.- СПб.: БХВ-Петербург, 2008.- 469с.
- 4) Информатика и программирование: компьютерный практикум: учеб.пособие: рек УМО/ А.Н.Гуда и др.; под общ.ред. В.И.Колесникова. – М.: Дашков и К, 2009, 2010.- 234с.

*Выводы по теме:*

Программное обеспечение компьютера – это совокупность программ, предназначенных для выполнения различных действий. В состав программного обеспечения включают программы и необходимые для их функционирования данные. Различают системные программы, предназначенные для управления и обслуживания компьютера и несистемные – программы (приложения). Все программы состоят из совокупности операторов и данных, описанных на некотором языке программирования, и создаются с помощью инструментальных программ. Все программы хранятся в файлах в виде либо текста программы на определенном языке программирования, либо в виде исполняемой программы. В первом случае для выполнения программы необходимо наличие транслятора или соответствующей системы программирования. Во втором случае, для выполнения программы достаточно просто запустить ее. Программное обеспечение принято классифицировать на три группы: системное; прикладное; инструментальное.

Основой системного программного обеспечения является *операционная система* (ОС), предназначенная для управления аппаратными и программными ресурсами компьютера, а также для организации взаимодействия (интерфейса) пользователя с компьютером. Операционная система представляет собой набор программ, хранимых в виде файлов на диске. Она автоматически загружается в оперативную память при включении и остается там до выключения компьютера. Эта операция загрузки выполняется загрузчиком- программой, которая вызывается базовой системой ввода-вывода (BIOS). BIOS размещается в постоянном запоминающем устройстве (ПЗУ), к которому доступ пользователя запрещен. Кроме вызова программы загрузчика BIOS также выполняет тестирование основных ап-

паратных компонентов. BIOS иногда относят к аппаратным средствам, иногда к программным.

Все современные дисковые операционные системы обеспечивают создание *файловой системы*, предназначенной для хранения данных на дисках и обеспечения доступа к ним.

Принцип организации файловой системы — табличный. Поверхность жесткого диска рассматривается как трехмерная матрица, измерениями которой являются номера поверхности, цилиндра и сектора. Под цилиндром понимается совокупность всех дорожек, принадлежащих разным поверхностям и находящихся на равном удалении от оси вращения. Данные о том, в каком месте диска записан тот или иной файл, хранятся в системной области диска в специальных таблицах размещения файлов (FAT-таблицах). Поскольку нарушение FAT-таблицы приводит к невозможности воспользоваться данными, записанными на диске, к ней предъявляются особые требования надежности, и она существует в двух экземплярах, идентичность которых регулярно контролируется средствами операционной системы.

#### *Создание и именование файлов*

*Файл* — это именованная последовательность байтов произвольной длины. Поскольку из этого определения вытекает, что файл может иметь нулевую длину, то фактически создание файла состоит в присвоении ему имени и регистрации его в файловой системе — это одна из функций операционной системы. Даже когда мы создаем файл, работая в какой-то прикладной программе, в общем случае для этой операции привлекаются средства операционной системы.

#### *Создание каталогов (папок)*

*Каталоги (папки)* — важные элементы иерархической структуры, необходимые для обеспечения удобного доступа к файлам, если файлов на носителе слишком много, то файлы объединяются в каталоги по любому общему признаку, заданному их создателем (по типу, по принадлежности, по назначению, по времени создания и т. п.). Каталоги низких уровней вкладываются в каталоги более высоких уровней и являются для них вложенными. Верхним уровнем вложенности иерархической структуры является корневой каталог диска.

#### *Копирование и перемещение файлов*

В неграфических операционных системах операции копирования и перемещения файлов выполняются вводом прямой команды в поле командной строки. При этом указывается имя команды, путь доступа к каталогу-источнику и путь доступа к каталогу-приемнику.

#### *Удаление файлов и каталогов (папок)*

Средства удаления данных не менее важны для операционной системы, чем средства их создания, поскольку ни один носитель данных не обладает бесконечной емкостью. Существует как минимум три режима удаления данных: удаление, уничтожение и стирание, хотя операционные системы обеспечивают только два первых режима (режим надежного стирания данных можно обеспечить лишь специальными программными средствами).

#### *Навигация по файловой структуре*

Навигация по файловой структуре является одной из наиболее используемых функций операционной системы. Удобство этой операции часто воспринимают как удобство работы с операционной системой. В операционных системах, имеющих интерфейс командной строки, навигацию осуществляют путем ввода команд перехода с диска на диск или из каталога в каталог. В связи с крайним неудобством такой навигации, широкое применение нашли специальные служебные программы, называемые файловыми оболочками.

*Прикладное программное обеспечение* используется для решения задач определенной прикладной области. В качестве примеров можно привести системы тестирования



знаний, системы автоматизации бухгалтерских расчетов, системы мониторинга, системы анализа эффективности инвестиций, системы документооборота и др. Разработка таких систем выполняется в несколько этапов и осуществляется на основе инструментального программного обеспечения.

*Пакеты прикладных подпрограмм* – это совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих определенным единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм и т. п.

#### Тема 7. Обзор языков высокого уровня (6 часов)

Цель: раскрыть сущность и применение языков высокого уровня

План:

- 1) Концепция типа языка ПАСКАЛЬ
- 2) Логический тип
- 3) Символьный тип (Char) \
- 4) Адресный тип (Pointer)

*Литература:*

Основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек. Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

Дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.
- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапорев С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапорев.- СПб.: БХВ-Петербург, 2008.- 469с.

*Выводы по теме:*

Языки программирования высокого уровня – это формальные языки специально созданные для общения человека с компьютером. Каждый язык программирования, равно как и "естественный язык" (русский, английский и т.д.) имеет алфавит, словарный запас, свои грамматику и синтаксис, а также семантику.

*Алфавит* – фиксированный для данного языка набор основных символов, допускаемых для составления текста программы на этом языке.

*Синтаксис* – система правил, определяющих допустимые конструкции языка программирования.

*Семантика* – система правил однозначного толкования отдельных языковых конструкций, позволяющих воспроизвести процесс обработки данных.

При описании языка и его применении используют *понятия языка*. Понятие подразумевает некоторую синтаксическую конструкцию и определяемые ею свойства программных объектов или процесса обработки данных.

Для того, чтобы составить программу на языке Турбо Паскаль надо знать структуру программы и правила использования в программах различных данных. Самое общее правило гласит : все данные, используемые Вами, должны быть предварительно определены, т.е. Вам необходимо задуматься о том, как обозначить каждое данное (какое дать ему имя), о характере и диапазонах изменений их значений, о требуемой памяти для их размещения и о наборе допустимых к ним операций.

Для определения данных Турбо Паскаль предлагает использовать способы объявления данных в программе посредством задания их значений или путем задания их типов - для данных изменяющих свои значения при выполнении программы.

Концепция типа языка ПАСКАЛЬ имеет следующие основные свойства:

1. любой тип данных определяет множество значений, к которому принадлежит константа, которые может принимать переменная или выражение, или вырабатывать операция или функция; тип значения, задаваемого константой, переменной или выражением, можно определить по их виду или описанию;

2. каждая операция или функция требует аргументов фиксированного типа и выдает результат фиксированного типа. Отсюда следует, что транслятор может использовать информацию о типах для проверки вычислимости и правильности различных конструкций.

Тип определяет:

- возможные значения переменных, констант, функций, выражений, принадлежащих к данному типу;
- внутреннюю форму представления данных в ЭВМ;
- операции и функции, которые могут выполняться над величинами, принадлежащими к данному типу.

В ТУРБО ПАСКАЛЬ ограничения на совместимость типов можно обойти помощью приведения типов. Приведение типов позволяет рассматривать одну и ту же величину в памяти ЭВМ как принадлежащую разным типам.

*Логический тип* (Boolean) определяет те данные, которые могут принимать логические значения TRUE и FALSE.

*Символьный тип* (Char) определяет упорядоченную совокупность символов, допустимых в данной ЭВМ. Значение символьной переменной или константы - это один символ из допустимого набора.

*Адресный тип* (Pointer) определяет переменные, которые могут содержать значения адресов данных или фрагментов программы. Для хранения адреса требуются два слова (4 байта), одно из них определяет сегмент, второе - смещение.

*Массивы* представляют собой ограниченную упорядоченную совокупность одно-типных величин. Каждая отдельная величина называется компонентом массива. Тип компонент может быть любым, принятым в языке ПАСКАЛЬ, кроме файлового типа. Тип компонент называется базовым типом. Вся совокупность компонент определяется одним именем. Для обозначения отдельных компонент используется конструкция, называемая переменной с индексом или с индексами:

Для ввода и вывода данных используются стандартные процедуры ввода и вывода Read и Write, оперирующие стандартными последовательными файлами INPUT и OUTPUT.

Вывод каждого значения в строку экрана происходит в соответствии с шириной поля вывода, определяемой конкретной реализацией языка. Форма представления значений в поле вывода соответствует типу переменных и выражений: величины целого типа выводятся как целые десятичные числа, действительного типа - как действительные десятичные числа с десятичным порядком, символьного типа и строки - в виде символов, логического типа - в виде логических констант TRUE и FALSE.

*Условный оператор* позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. Таким образом, условный оператор - это средство ветвления вычислительного процесса.

*Отличия двух операторов цикла с условиями.*

- В операторе цикла с предусловием возможна такая ситуация, что операторы, содержащиеся внутри него не будут выполнены ни одного раза, т.к. условие выхода из цикла там проверяется перед выполнением оператора. В операторе с постусловием оператор, содержащийся в цикле выполняется хотя бы один раз.

- В этих двух операторах по-разному проверяется логическое выражение на выход из цикла.

*Выход из цикла.*

В циклах с условиями при большом объеме вычислений возможна ситуация, когда время выполнения цикла становится слишком большим. В этом случае необходимо использовать стандартную процедуру прекращения выполнения цикла **BREAK**.

Эта процедура используется т.ж. для досрочного выхода из цикла в случае получения результата до конца выполнения цикла.

Процедура **CONTINUE** используется для досрочного перехода к следующей операции цикла.

*Двумерный массив* – это структура однотипных элементов, расположенных в виде таблицы значений.

Такое представление значений соответствует математическому понятию двумерный массив. Каждый элемент в двумерном массиве идентифицируется *номером строки и номером столбца*, на пересечении которых он расположен. Например, в двумерном массиве А, изображенном на рис. 24, элемент со значением 5 расположен на пересечении третьей строки и второго столбца. Этот элемент будет обозначаться как А(3,2). А элемент А(1,4) имеет значение, равное нулю. Такое представление набора значений позволяет выполнять обработку как отдельных значений в двумерном массиве, так и последовательности значений, расположенных в строках или столбцах. В дальнейшем будем считать, что для двумерного массива А(N,M) в обозначении элемента А(i,j) первое значение i соответствует номеру строки и изменяется от 1 до N, а j - номеру столбца и изменяется от 1 до M. В отличие от одномерного массива, в котором использовался только один номер для определения местоположения элемента и требовался только один цикл для ввода элементов, в двумерном массиве для обработки элементов необходимы два вложенных друг в друга цикла. Внешний цикл предназначен для изменения номера строки i, а второй, внутренний, - для изменения номера столбца j в текущей строке i.

Подпрограммы являются одним из основных средств структурирования программ, то есть представления программы в виде отдельных синтаксических конструкций, имеющих уникальное имя, и предназначенных для выполнения определенного алгоритма.

Применение подпрограмм дает возможность уменьшить число повторений одной и той же последовательности операторов, а также конструировать программу как набор отдельных подпрограмм. Это позволяет получить более логичный процесс программирования. Каждая процедура или функция определяется только один раз, но может использоваться многократно.

*Процедурами* в Турбо Паскале называют вид подпрограмм. Они предназначены для оформления любой самостоятельной части программы в виде отдельной синтаксической конструкции, которая решает определенную задачу. Процедуры используются для изменения окружающей среды программы и, как правило, определяют некоторые действия.

*Подпрограмма-функция* предназначена для вычисления какого-либо одного значения. Описание каждой функции начинается с заголовка, в котором задаются имя функции, список формальных параметров с указанием их типов и тип значения функции.

С помощью параметров осуществляется передача исходных данных в подпрограмму.

*Формальные параметры* указываются в заголовке подпрограммы, *фактические параметры* – при обращении к подпрограмме.

Само описание процедуры не вызывает конкретных действий. Для того, чтобы выполнить действия, указанные в процедуре, необходимо в вашей программе использовать оператор вызова соответствующей процедуры. Например, для вызова процедуры в нужном месте программы указывается имя этой процедуры, за которым в круглых скобках могут следовать параметры, передаваемые из программы в процедуру. Такие параметры называются *фактическими*.

Все *формальные параметры* можно разбить на четыре категории:

1) *параметры-значения* (они в основной программе подпрограммой не меняются);

- 2) параметры-переменные (их подпрограмма может изменить в основной программе);
- 3) параметры-константы (используются только в версии 7.0);
- 4) параметры без типа.

Для каждого формального параметра следует указать имя и, как правило, тип. Имена параметров могут быть любыми, в том числе и совпадать с именами объектов программы. Необходимо лишь помнить, что в этом случае параметр основной программы с таким именем становится недоступным для непосредственного использования подпрограммой. Тип формального параметра может быть практически любым, однако в заголовке подпрограммы нельзя использовать структурный тип, например, описание массива или записи.

#### Тема 8. Технология программирования (2 часа)

*Цель:* рассмотреть технологию программирования

*План:*

- 1) Жизненный цикл программ
- 2) Основные принципы структурной методологии
- 3) Методы проектирования "водопадная" модель, "спиральная" модель.

*Литература:*

Основная литература:

- 1) Информатика. Базовый курс: учебное пособие: рек. Мин. Обр. РФ/ под редакцией Симоновича С.В. – СПб: Питер Пресс, 2007, 2009. - 640 с.
- 2) Информатика: учеб.: рек. Мин. Обр. РФ/ под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика, 2005, 2007, 2009. – 768 с.
- 3) Острейковский В.А. Информатика [Текст]: учеб.: рек. Мин. обр. РФ / В.А.Острейковский. – М.: Высш. Шк., 2009. – 512 с.

Дополнительная литература:

- 1) Кацко И.А. Практикум по анализу данных на компьютере: учеб.-практ. пособие: рек. УМО/ И.А.Кацко, Н.Б. Паклин. – М.: КолосС. 2009.- 279с.
- 2) Информатика [Текст]: учеб. / Б.В.Соболь и др. – Ростов н/Дону: Феникс, 2009. – 447 с.
- 3) Шапорев С.Д. Информатика. Теоретический курс и практические занятия : учеб. пособие: рек. НМС/ С.Д. Шапорев.- СПб.: БХВ-Петербург, 2008.- 469с.

*Выводы по теме:*

Традиционная технология программирования формировалась на заре вычислительной техники, когда в распоряжении пользователей были ограниченные ресурсы ЭВМ, а разработчик программ был в то же время и главным ее пользователем. В этих условиях главное внимание обращалось на получение эффективных программ в смысле оптимального использования ресурсов ЭВМ.

Периодически повторяющийся процесс разработки, внедрения и эволюции программного продукта принято называть его *жизненным циклом*. Жизненный цикл программ состоит из нескольких *этапов*. Современные программные системы относятся к классу больших и сложных систем. Один разработчик не в состоянии охватить все аспекты ее функционирования. Для решения этой проблемы целесообразно использовать следующие стратегические подходы:

- анализ сложности задачи
- абстракция (вместо сложного реального объекта рассматривается обобщенная модель)
- декомпозиция – разделение большой системы на меньшие подсистемы, которые вновь разделяются на меньшие составляющие.
- построение иерархии выделенных компонент.

Программы небольшого и среднего размера (несколько тысяч строк) создаются, как правило в несколько этапов. Сначала необходимо точно *определить требования*,

предъявляемые заказчиком к программному продукту. Этот этап очень важен, так как от правильного понимания целей и требований зависит в основном успех реализации.

На следующем этапе выбирается *методология разработки ПО*, технология программирования, система программирования и выполняются работы по созданию программного продукта..

*Этап внедрения..* Этот этап обычно заканчивается подписанием акта приемки работ по созданию программы.

Последним этапом является *этап сопровождения программы*. На этом этапе создатели программы гарантируют на определенный период бесплатное разрешение вопросов, возникающих при эксплуатации программы, а также информируют заказчика о новых усовершенствованных версиях программы.

Процесс разработки программного продукта требует определенной организации. Для этого были разработаны *методы проектирования*. Наиболее распространенными моделями процесса разработки программ являются:

- "водопадная" модель;
- "спиральная" модель.

Особенность "*спиральной модели*" заключается в том, что "степень готовности" продукта считается возрастающей величиной в зависимости от времени и этапа разработки. Функция возрастания "степени готовности" изображается в полярной системе координат в виде спирали. "Спиральная" модель по сравнению с "водопадной" моделью имеет *достоинства*:

- предусмотрено быстрое прототипирование будущей программы
- предусмотрена разработка программ с созданием последовательных версий.

*Структурное программирование* можно толковать как "проектирование, написание и тестирование программы в соответствии с заранее определенной дисциплиной".

К основным *методам структурного программирования* относится, прежде всего, отказ от бессистемного употребления оператора GOTO и преимущественное использование других структурированных операторов, методы нисходящего проектирования разработки программы, идеи пошаговой детализации и некоторые другие соглашения, касающиеся дисциплины программирования.

Всякая программа, в соответствии с структурным подходом к программированию, может быть построена только с использованием трех основных типов блоков.

1. *Функциональный блок*, который на блок-схеме изображается в виде прямоугольников с одним входом и одним выходом:

2. *Условная конструкция*. Этот блок включает проверку некоторого логического условия (P), в зависимости от которого выполняется либо один (S<sub>1</sub>), либо другой (S<sub>2</sub>) операторы:

3. *Блок обобщенного цикла*. Этот блок обеспечивает многократное повторение выполнения оператора S пока выполнено логическое условие P:

#### Основные принципы структурной методологии

1. *Принцип абстракции*. Этот принцип предполагает рассмотрение всей программной систем, как многоуровневые системы. Каждый уровень является детализацией предыдущих.

2. *Принцип формальностей*. Каждая программа должна реализовывать некоторый алгоритм, который построен на определенной математической модели решения задач. Математическая модель содержит математические зависимости и формы, а также ограничения, накладываемые на исходные данные. При этом формулы предназначаются для преобразования исходных данных в результат.

3. *Принцип "разделяй и властвуй"*. Определяет способ решения трудных задач путем разделения этой задачи на множество мелких, легко решаемых. Этот принцип реализуется путем создания подпрограмм.

4. *Принцип модульности.* Определяет способ создания больших программ, у которых возникают проблемы с оперативной памятью, в следствие их большого объема или большого объема обрабатываемых ими данных. Принцип модульности ускоряет создание больших программ за счет использования ранее созданных описаний.

5. *Принцип открытости.* Программы должны быть открытыми для быстрых модификаций, поэтому они должны быть понятны и хорошо откомментированы. Использование структурной методологии позволило повысить надежность и эффективность программ, а также сократить сроки создания и сопровождения программ.

### **3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ (РЕКОМЕНДАЦИИ)**

#### **3.1 Методические указания (рекомендации) преподавателя**

Методические рекомендации по изучению дисциплины представляют собой комплекс рекомендаций и разъяснений, позволяющих студентам оптимальным образом организовать процесс изучения данной дисциплины.

*Работа с учебниками.* При работе с книгами рекомендуется делать конспективные записи, а не копировать текст, например, посредством сканера. То есть работа с источниками – это не только чтение, но и выписывание нужного материала, что повышает уровень усвоения и запоминания.

*Работа с конспектом лекций.* Знакомство с университетской системой образования происходит уже на первой лекции, где от студента требуется не просто внимание, но и самостоятельное оформление конспектов. Чтобы грамотно его составить, а затем с максимальной пользой использовать, студенту необходимо знать о видах читаемых лекций. Вообще «лекция» представляет собой систематическое, последовательное изложение учебного материала, какого-либо вопроса, темы, раздела. Обычный вузовский курс, в котором последовательно излагается материал по учебной программе, включает следующие виды лекций:

- вводная лекция, где преподаватель дает общие представления о дисциплине, ее предмете и объекте, определяет цели и задачи курса, методологию и методы, рекомендует литературу, дает ее анализ;

- текущие лекции по конкретным темам курса, которые также разделяются на виды. Преподаватель может просто ознакомить с новой темой, выделить основные моменты, объяснить причинно-следственные связи, сделать выводы, - это обычный вариант лекции. Как правило, она не вызывает затруднений в конспектировании.

Вместе с тем может быть прочитана «проблемная» лекция по какому-либо дискуссионному вопросу, на которой приводятся точки зрения и аргументы различных ученых, дается их критический анализ. Это более сложный вариант лекции, так как предполагается, что студенты уже владеют фактическим материалом и основными понятиями. Поэтому без усвоения уже пройденного материала студенту будет сложно понять обсуждаемую проблему. Это, в свою очередь, не позволит студенту правильно законспектировать лекцию и затем использовать записи при подготовке к экзамену.

Заключительная лекция, в которой делаются общие выводы по прочитанной дисциплине, характеризуются итогами и результатами, определяются тенденции, анализируются перспективы.

Таким образом, при работе с конспектом лекций нужно учитывать ее разновидность. Одни лекции дают ответы на конкретные вопросы темы, другие – лишь выявляют взаимосвязь между явлениями, помогая понять глубинные вопросы.

Несмотря на наличие разных видов лекций существует несколько общих правил по их конспектированию и дальнейшей работе с записями.

#### **3.2 Методические указания к лабораторным работам**

##### **Лабораторная работа № 1- Устройство компьютера. работа с окнами windows**

*Цель:* изучить устройство компьютера, работу с окнами windows

*План:*

- 1) Устройство компьютера

- 2) Вид экрана и окна
- 3) Изменение размера окна
- 4) Работа с простыми базовыми программами

#### *Методические указания*

Компьютер состоит из трех частей:

- системного блока;
  - клавиатуры, позволяющей вводить символы в компьютер;
  - монитора (дисплея) – для изображения текстовой или графической информации
- В системном блоке располагаются основные узлы компьютера:
- электронные схемы, управляющие работой компьютера (микропроцессор, оперативная память, контролеры устройств и т.д.);
  - блок питания, преобразующие электропитание сети в постоянный ток низкого напряжения, подаваемый на электронные схемы компьютера;
  - накопители (дисководы) для гибких магнитных дисков, используемые для чтения и записи на гибкие магнитные диски (дискеты);
  - накопители на жестком магнитном диске, предназначенные для чтения и записи на несъемный жесткий магнитный диск (винчестер).

#### 2. Вид экрана.

После загрузки Windows большую часть экрана занимает Рабочий стол, причем он выглядит так же, как поверхность обычного стола. На нем можно «разложить» нужные программы. Настройка рабочего стола включает в себя размещение на столе пиктограмм (**пиктограмма – значок ускоренного доступа к файлу или программе**), изменение параметров фона рабочего стола, рисунков заставок и другие действия. Быстрый запуск программ с рабочего стола производится установкой на пиктограмму нужной программы курсора с помощью мыши и двойным щелчком ее левой кнопки.

В нижней части экрана расположена **панель задач**. Правая часть панели используется для системных индикаторов (время, русская или английская раскладка клавиатуры и др.) Центральная часть панели задач используется для отображения данных об активных программах. Если запущена какая-либо программа, то на панели задач появляется пиктограмма данной программы и ее название. Нажав мышью на кнопку пиктограммы на панели задач, можно перейти в данную программу (окно программы выйдет на первый план и будет доступно для работы). В левой части панели расположена кнопка **«Пуск»**, после нажатия на которую раскроется **главное меню**. При выборе какой-либо из строк, заканчивающейся знаком треугольника, открывается дополнительное окно, называемое **меню второго уровня** и т.д.

На рабочем столе обязательно должна находиться системная папка **«Мой компьютер»**, в которой отражается содержание компьютера целиком.

**Корзина** – это специальная системная папка, в которую помещаются удаленные файлы. Ее значок находится на Рабочем столе. Любой файл можно восстановить на прежнее место, на котором он находился перед удалением. Файл безвозвратно теряется, если он удаляется из корзины. Удалить все файлы из корзины можно, выполнив команду «очистить корзину». Чрезмерное заполнение корзины может привести к нехватке дисковой памяти.

#### 3 Вид окна

В правом верхнем углу прикладной программы находятся кнопки увеличения (уменьшения), минимизация и закрытия окна программы.

Когда Вы нажимаете кнопку минимизации, прикладная программа останется в памяти системы, а окно сворачивается. Для возвращения в программу необходимо выбрать ее пиктограмму на панели задач.

В верхней части окна располагается **строка заголовка**, в которой выводится название прикладной программы и имя открытого файла.

Под строкой заголовка располагается **строка меню**. Нажимая на клавишу, соответствующую той или иной операции, можно быстро выполнить ее без поиска нужной команды.

Ниже строки меню располагается **панель инструментов**, с помощью которой можно быстро задать команду для форматирования текста.

Если часть информации не помещается в окне, то на ее правой и нижней границах находятся **полосы прокрутки**. Они позволяют просмотреть все содержимое документа. Оба конца полосы прокрутки оканчиваются стрелкой. Этими стрелками можно плавно перемещать бегунок – прямоугольник на полосе прокрутки. Его положение отмечает часть документа, которая в данный момент отображается на экране. Перетаскивая бегунок мышкой, можно просмотреть содержимое окна гораздо быстрее, чем с помощью полосы прокрутки.

4. Изменение размеров окна и перемещение его по экрану.

Если подвести курсор мыши к границе окна, то он примет вид двунаправленной стрелки. Нажав на клавишу мыши и передвигая ее, можно изменить размеры окна в направлении стрелки.

Для перемещения окна по экрану следует поместить курсор на строку заголовка, нажать клавишу мыши и, не отпуская ее, переместить мышь вместе с окном в нужное место.

Если открыты сразу несколько программ, то текущее окно (программа) – самое верхнее. Оно перекрывает остальные окна. Строка заголовка у текущего окна цветная, у других – нет.

Чтобы изменить одновременно вертикальные и горизонтальные размеры окна, надо подвести мышь к любому углу окна, нажать левую кнопку мыши и, не отпуская, изменить размеры окна до нужных размеров.

Открытые окна могут быть расположены на экране каскадом или мозаикой. Для расположения окон одним из указанных способов следует поместить курсор мыши на свободный участок панели задач и щелкнуть правой клавишей мыши. В появившемся меню выбрать соответствующую опцию.

5 Стандартные программы

*Калькулятор*. Принцип работы с калькулятором аналогичен обычным калькуляторам. Возможны два варианта работы с калькулятором: обычный и научный. Если возникают сомнения в назначении клавиш, то следует нажать над клавишей правую кнопку мыши и щелкнуть по появившейся надписи «Что это?» правой клавишей мыши. На экране появится необходимое описание.

*Блокнот, Word Pad*. Блокнот – то простейший текстовый не форматлируемый редактор. Word Pad предназначен для работы с документами с необходимым оформлением.

Блокнот - это простейший текстовый не форматлирующий редактор. Интересной возможностью является автоматическое добавление в текст редактируемого документа текущего времени и даты. Для этого необходимо в первой строке документа с первой позиции вставить команду .LOG (обязательно прописными буквами). После этого при каждом открытии документа в конец текста будет добавляться строка с текущей датой.

Текстовый процессор WordPad предназначен для создания документов с необходимым оформлением. По своим возможностям уступает редактору WinWord.

Задание 1:

1. Загрузить программу блокнот.
2. Создать файл, содержащий текст на русском и английском языках.
3. Сохранить файл, задав ему имя.
4. Изучить меню программы Блокнот.
5. Ввести команду для автоматического добавление в текст редактируемого документа текущего времени и даты.

Задание 2:



1. Загрузить текстовый процессор WordPad.
2. Создать второй файл.
3. Изучить возможности программы.

Задание 3:

1. Изучить элементы окна программы: меню, изображения инструментов, палитру цветов.
2. Открыть какой-либо из созданных ранее файлов (если таковой имеется).
3. Создать новый файл.
4. Выполнить рисунок, представить его в различных масштабах, проработать все пункты меню.
5. Выйти из программы Paint.
6. Открыть окна созданных Вами файлов.
7. Задать открытым окнам размеры и местоположение.
8. Разместить окна соответствующих программ на экране различными способами.
9. Закрыть все окна.

*Задания для самостоятельной работы.*

1 Разместите пиктограммы, находящиеся на Рабочем столе, в произвольном порядке. При помощи контекстного меню Рабочего стола упорядочьте их размещение.

2 Установите новые параметры Рабочего стола (фон, заставка, оформление) воспользовавшись диалоговыми окнами. После установки новых параметров верните интерфейс в исходное положение.

3 Откройте главное меню. Выберите меню «Программы». Перейдите в подменю Программы.

4 В подменю Программы выберите пункт Стандартные. Перейдите в подменю Стандартные.

5 Откройте программу Блокнот. Напишите любой текст.

6 Не закрывая предыдущую программу откройте программу Word Pad и нарисуйте поздравительную открытку.

7 Переключитесь в программу Блокнот, используя панель задач.

8 Уменьшите размеры окна Блокнот с помощью кнопки «уменьшить» в строке заголовка.

9 Переключитесь в программу Word Pad с помощью мыши.

10 Не закрывая программы, откройте программу Калькулятор. Выполните пример расчета по заданию преподавателя.

11 Расставьте окна программ каскадом, сверху вниз.

**Лабораторная работа № 3 - Работа с файлами и каталогами (папками)**

*Цель:* создание каталогов и папок, работа по определению состояния файлов и папок

*План:*

- 1) Создание каталогов и папок
- 2) Методы переноса папок
- 3) Методы копирования папок
- 4) Удаление и перенос папок и файлов
- 5) Нахождение информации о папках и файлах

*Методические указания*

*1.1 Создание каталога и файла на рабочем столе*

- наведите указатель мыши на свободное место рабочего стола и щелкните правой кнопкой мыши;

- в открывшемся меню выберите пункт «Создать» и в появившемся меню щелкните на пункт «Папка» или «Текстовый документ»;

- наберите имя каталога или файла и нажмите клавишу «Enter»

*Замечания:*

- в WINDOWS каталоги называются папками;
- имя каталога или файла может быть записано как русскими, так и латинскими буквами.

Созданные таким образом папка или файл окажутся расположенными в каталоге «Рабочий стол». Их можно увидеть, раскрыв этот каталог с помощью программы «Проводник» (см. пункт «Программы» главного меню или с помощью строки меню). Для помещения созданной папки в нужный каталог необходимо произвести с ней одну из предложенных операций:

- создание папки сразу в заданном каталоге;
- перенос в заданный каталог.

*1.3. Создание папки или файла в заданном каталоге*

Откройте окно «Проводник» с помощью пункта «Программы» главного меню;

В левой части окна «Проводник» щелкните мышью на папке, в которой нужно создать папку или файл. Папка раскроется, и ее содержимое отразится в правой части окна;

В правой части окна щелкните правой кнопкой мыши;

Создайте каталог или файл аналогично пункту 1.

*2. Перенос папки (файла)*

*1-й способ*

- найти папку(файл), который нужно перенести;
- открыть папку, куда нужно осуществить перенос;
- «возьмите» правой кнопкой мыши папку (файл) и, не отпуская, тяните ее к папке, куда осуществляется перенос. Вместе с указателем перемещается и пиктограмма каталога (файла);

- когда перемещающая папка (файл) выделится, отпустите кнопку мыши. В результате на старом месте каталог (файл) исчезнет;

Просмотрите «принимающую папку, раскрыв ее двойным щелчком мыши.

*2-й способ*

- найдите папку (файл), который нужно перенести;
- щелкните правой кнопкой мыши, в контекстном меню выберите пункт «вырезать» и щелкните левой кнопкой мыши;

- откройте папку, куда нужно осуществить перенос;

- щелкните правой кнопкой мыши. В контекстном меню выберите пункт «вставить» и нажмите левую кнопку. При этом переносимая папка (файл) окажется в нужной папке, но на старом месте ее не будет.

*3. Копирование папки (файла)*

*1-й способ*

Выполняется аналогично переносу

*2-й способ*

- наведите курсор мыши на нужную папку (файл) и нажмите правую кнопку;
- в контекстном меню выберите пункт «копировать» и нажмите левую кнопку мыши»

- войдите в папку, куда нужно копировать. Нажмите в рабочей области окна правой кнопкой мыши и выберите команду «вставить».

*4. Переименование папки (файла)*

*1-й способ*

- щелкните на имени папки (файла) левой кнопкой мыши. Имя выделится. Подождите секунду и щелкните на имени еще раз. Имя обведется в рамку;

- введите новое имя, зафиксируйте его клавишей «Enter» или мышью на свободном месте рабочего стола

*2-й способ*

Переименовать папку (файл) можно, указав на нее мышью, щелкнув правой кнопкой и выбрав в открывшемся меню пункт «переименовать».

#### 5. Удаление папки (файла)

- найдите нужный файл;

удаление через меню:

- укажите курсором на имя папки (файла), щелкните правой клавишей;
- выберите пункт меню «Удалить»;
- убедитесь, что удаляете нужный файл (папку), и подтвердите удаление.

удаление вручную

- укажите мышью на пиктограмму файла, нажмите на левую клавишу;
- не отпуская клавишу, перетащите файл в «Корзину»;
- отпустите клавишу мыши.

#### 6. Восстановление удаленного файла

- откройте содержимое корзины двойным щелчком левой кнопки мыши;- найдите нужный файл и, выделив его имя, перетащите на рабочий стол. Тоже можно выполнить через меню «файл/восстановить»;

- если необходимо, перенесите файл в нужную папку.

#### 7. Получение информации о каталоге

- откройте папку о которой нужно получить информацию;  
- через меню «Вид/Упорядочить значки» или с помощью контекстного меню (щелкните правой кнопкой мыши на свободном месте рабочего поля) выберите удобный вариант представления содержимого каталога:

- «по имени» - для расстановки по алфавиту имен;
- «по типу» - для расстановки файлов различных типов группами;
- «по размеру» - в порядке увеличения объема занимаемой памяти;
- «по дате» - устанавливает первыми самые последние из созданных файлов.

Выберите последовательно каждый пункт и просмотрите изменения.

#### 8. Поиск папок и файлов по разным характеристикам

В главном меню выберите пункт «Поиск/папки и файлы»

##### 8.1. поиск по имени

Осуществляется через пункт «Имя и размещение»;

- в строке «Часть имени» запишите любую часть имени файла (не обязательно с начала имени) для поиска всех папок и файлов с указанным набором символов;  
- нажмите кнопку «найти».

При написании имени файла можно использовать символы (\*) и (?). Символ (\*) соответствует любому количеству букв, а символ (?) – одной букве.

##### 8.2 Поиск по фрагменту информации

Осуществляется через пункт «Слово или фраза в файле»

- в строке «Слово или фраза в файле» наберите известный вам фрагмент текста, по которому можно найти файл;  
- нажмите кнопку «найти».

##### 8.3. Поиск по дате создания или последнего изменения

Осуществляется через пункт «когда были произведены последние изменения»:

- выберите один из вариантов ограничения поиска;
- нажмите кнопку «Пуск».

##### 8.4. Поиск по заданному размеру занимаемой памяти

Осуществляется через пункт «Указать диапазон»:

- задайте границы размера в Кбайтах в строке «Размер»;
- нажмите кнопку «найти»

При переходе на новый этап поиска не забывайте обновлять  
(правильно устанавливать) исходные параметры поиска!!!

## Задания для самостоятельной работы № 2

1 В корневом каталоге C (на диске C) создать каталог «STUDENT»

2 В каталоге «STUDENT» создать каталог «GRUPP»

3 В каталоге «STUDENT» создать два текстовых файла:

**ЛАБ.1**, содержащий следующий текст «После окончания школы № \_\_\_\_ я поступил учиться в Амурский государственный университет на специальность «\_\_\_\_\_». После окончания университета я буду инженером в области \_\_\_\_\_».

**ЛАБ.2**, содержащий следующий текст: «В нашей группе \_\_ человек. Все мы обязательно сдадим первую сессию на «хорошо» и «отлично». Наша группа – самая лучшая на факультете.»

4 В каталоге «GRUPP» создать каталог **DISK**

5 В каталоге **DISK** создать два текстовых файла:

**КАТАЛОГ**, содержащий следующий текст: «Текущим называется каталог, в котором в данный момент работает пользователь»

**ПУТЬ**, содержащий текст: «Путь – это последовательность имен каталогов, разделенных символом «\». Этот путь задает маршрут к тому каталогу, где находится нужный файл»

6 В каталоге «STUDENT» создайте файл «ИТОГ», в который скопируйте содержимое файлов **КАТАЛОГ** и **ПУТЬ**.

7 Скопировать файл **ЛАБ.1** в каталог **GRUPP**

8 Просмотреть содержимое папки **STUDENT** с помощью «Проводника». Записать в тетради дерево папки «STUDENT»

Найти файлы по имени:

- 1) ????.exe
- 2) W\*.DDL
- 3) Все файлы с именем, не более чем из трех символов
- 4) Все файлы с расширением sys
- 5) Все файлы с именем, начинающимся на S и состоящим не более, чем из пяти символов
- 6) Все файлы, начинающиеся с символа N, имеющие в расширении три символа; последний символ t
- 7) Найдите все файлы и каталоги, имеющие в своем имени символы «0». Войдите в один из файлов и определите его размер занимаемой памяти
- 8) Сколько найдено файлов в каждом случае? Запишите результаты поиска в тетрадь

Откройте папку (по заданию преподавателя), находящуюся на диске C:

- задайте содержимое папки в виде списка, мелких значков.
- расставьте содержимое папки по алфавиту, убедитесь в этом;
- расставьте содержимое папки по размеру занимаемой памяти. Определите объем самого маленького файла;
- расставьте записи папки по дате создания. Сколько файлов создано вчера?

Найти файлы, созданные:

- за последнюю неделю
- за последний день

Показать результаты работы двух заданий преподавателю

Удалите все созданные Вами файлы и каталоги.

## Лабораторная работа № 6 - OPENOFFICE.ORG, форматирование текста

*Цель:* получить навыки работы в OPENOFFICE.ORG.

*План:*

- 1) Понятие об OPENOFFICE.ORG
- 2) Особенности форматирования текста в OPENOFFICE.ORG

**1. OpenOffice.org** - один из немногих качественных офисных пакетов для работы с текстовыми документами, электронными таблицами, изображениями, базами данных и презентациями. Пакет OpenOffice.org включает в себя следующие программы:

- OpenOffice.org Base - программа для создания небольших баз данных;
- OpenOffice.org Calc - программа для работы с электронными таблицами, аналог Microsoft Excel;
- OpenOffice.org Draw - программа для создания и редактирования изображений;
- OpenOffice.org Impress - программа для создания простеньких презентаций, аналог Microsoft PowerPoint;
- OpenOffice.org Writer - программа для работы с текстовыми документами, аналог Microsoft Word;

Каждая программа совместима со всеми широко распространенными форматами документов, электронных таблиц, презентаций и изображений.

## ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Напечатать предложенный текст

Перед форматированием текста (абзаца) его необходимо выделить

- 2 Отформатировать текст:

- установить поля страницы:
  - левое 3 см
  - правое – 1 см
  - верхнее – 2 см
  - нижнее – 2 см

Для этого необходимо войти в меню **Формат/Страница/Страница**

3 Озаглавить текст. Заголовок разместить посередине, размер шрифта – 16. Весь текст – шрифт 14. Записать в тетради возможные методы установления размера шрифта.

4. Установить начало абзацев с красной строки (1,30). Для этого в строке *меню* выбрать опцию **ФОРМАТ/АБЗАЦ /ОТСТУПЫ и ИНТЕРВАЛЫ**

5 Установить автоматический перенос слов. В строке *меню* выбрать опцию **ФОРМАТ/АБЗАЦ /ПОЛОЖЕНИЕ НА СТРАНИЦЕ**

6 Произвести выравнивание текст по ширине

7 Между заголовком и основным текстом установить интервал – 2. Основной текст печатается через интервал 1,5

8 Второй абзац скопировать в конец текста

9 Поменяйте третий абзац с четвертым.

10 Начиная с четвертого абзаца перенесите текст на другую страницу. Запишите в тетради способ переноса (как вы начинаете текст с новой страницы). Записать действия перехода на другую страницу в тетрадь

11 Текст должен быть напечатан шрифтом **Times New Roman**

12 В папке **МОИ ДОКУМЕНТЫ**, расположенной на диске **C**, создать папку со своими инициалами. Созданный файл сохранить под именем «**Лаб. 1**» и перенести в свою папку. Сегодня все более возрастают требования к одежде. Она должна быть красивой по форме, качественной по исполнению, а также функциональной, удобной в носке, не вредной для здоровья, в общем, дающей предельный комфорт и создающей определенный эстетический образ. Всем этим требованиям как нельзя лучше отвечает одежда из трикотажа.

Трикотажная одежда прочно вошла в обиход, основательно потеснив одежду из ткани. И теперь уже невозможно представить гардероб любого человека без трикотажных

изделий. Главная причина кроется в универсальности трикотажа: из него можно изготавливать практически все виды современной одежды, будь то белье, платье, костюм, пальто. Причем по многим параметрам трикотажная одежда не только не уступает тканевой, но и превосходит ее.

Если для потребителя трикотажная одежда не только приемлемая, но и зачастую и более удобная и красивая, чем одежда из ткани, то для производителя изготовление одежды из трикотажа значительно экономичнее и выгоднее, чем ее пошив из ткани. И прежде всего потому, что трикотажное производство менее трудоемкое, чем ткацкое. Одежда из трикотажа создается гораздо быстрее, чем из ткани, поскольку ее покрой и технология обработки намного проще; расход сырья и материалов на единицу продукции в трикотажном производстве тоже намного меньше.

В настоящее время очень высоки темпы развития трикотажного производства, что, с одной стороны, связано с ростом применения трикотажа для изготовления изделий, а с другой стороны, с успехами химической промышленности, обогатившей сырье различными химическими волокнами и нитями с самыми разнообразными свойствами. В связи с этим значительно возросли возможности художественного оформления одежды. Кроме традиционных видов оформления изделий появились дополнительные, позволяющие изменить фактуру, получать разнообразные внешние эффекты, а также придавать изделиям желаемые свойства.

### **Лабораторная работа № 8 - Технологии обработки графической информации**

**Цель:** научиться пользоваться инструментами панели **Рисование MS Word**, а также пользоваться графическим редактором **MS Visio**.

**План:**

- 1) С помощью инструментов панели **Рисование MS Word** научиться создавать автофигуры, добавлять текст в автофигуру,
- 2) группировка объектов, настройка объема и тени; и дополнительно: установление динамических связей между графическими объектами **MS Visio**.
- 3) Установить размеры автофигуры, добавить текст в автофигуру, сгруппировать несколько автофигур, изменять форматы созданных объектов.

#### *Методические рекомендации*

1. Создайте различные схемы, графики и рисунки с помощью инструментов панели **Рисование MS Word** и графического редактора **MS Visio**.

В данной лабораторной работе необходимо начертить с помощью инструментария **Рисование MS Word** и **MS Visio**. В графическом редакторе **MS Visio** следует начертить схемы 1, 3 и 4. В **MS Word** с помощью панели **Рисование MS Word** следует нарисовать все схемы. Содержание текстовых надписей может быть произвольным.

Приемы работы, панели инструментов и основные команды по работе с графическим редактором **MS Visio** приводятся в ПРИЛОЖЕНИИ 4.

В начале нарисуйте схему 1. Она состоит из нескольких объектов, соединенных между собой. Шрифт внутри каждого прямоугольника выберите 16 пунктов. Нарисованные объекты сгруппируйте.

Схема 2 позволит научиться создавать объекты с заливкой и с тенью. Выберите шрифт надписей 14 пунктов, курсивный.

Схему 3 нарисовать и в инструментарии **MS Word** и с помощью графического редактора **MS Visio**. Обратите внимание на объект **Линия**, который может иметь разные форматы (толщина, тип, цвет, наличие стрелок на концах). Обязательно сделайте привязки прямоугольников и других фигур ко всем примыкающим к ним линиям.

Схема 4 позволяет изобразить пример наиболее часто встречающихся при проектировании предприятий схем. Это схема производственного состава ателье. Нарисуйте ее также, как и предыдущие схемы в **Word** и **MS Visio**.

#### *Самостоятельная работа*

1. Выполните с помощью панели инструментов MS Word рисунок поздравительной открытки (Рис.2).

При выполнении работы рекомендуется использовать также приложение *MS Art* из меню **Вставка/ Объект**. Текст содержания открытки может быть произвольным. Количество объектов в рисунке не менее пяти. При добавлении новых объектов не забудьте про их группировку и порядок размещения – на передний план, задний план и т.д. -

Сохранить полученные документы, скопируйте на дискете 3,5 А, компакт-диске или на носителе флэш-памяти.



#### **Лабораторная работа № 14 - создание таблиц. работа с таблицами**

*Цель:* научиться работать с таблицами в текстовом редакторе, выполнять математические операции, форматировать таблицы, создавать таблицы с различным стилем.

*План:*

- 1) Вставка таблиц, добавление ячеек, строк;. Объединение ячеек, строк
- 2) Форматирование таблицы: установление размеров строк, столбцов
- 3) Создание различных стилей таблицы
- 4) Выполнение математических действий в таблице

#### *Методические рекомендации*

Таблица представляет собой некий объем информации, организованный с помощью строк и столбцов. Ячейки таблицы могут содержать не только текст, но и графические и прочие объекты.

Создать таблицу можно следующими способами:

- в строке меню выбрать команду **Таблица/Вставить/Таблица**. В появившемся окне указать количество столбцов и строк;
- в строке меню выбрать команду **Вставка/Таблица** и в появившемся окне задать количество строк и столбцов;
- щелкнуть кнопкой мыши на стандартной панели инструментов . В появившемся окне также выбрать число столбцов и строк, которое вам нужно.

Редактирование структуры таблицы сводится к следующим операциям:

- добавление заданного количества строк (столбцов);
- удаление выделенных ячеек, строк, столбцов;
- разбиение выделенных ячеек.





Для таблиц также существуют различные стили форматирования. Для выбора стилей необходимо выполнить команду **Таблица/Автоформат** и выбрать атрибут форматирования.

Обрамление таблицы. Ячейки таблицы могут быть обрамлены границами или линиями сетки. Линии сетки – это служебные линии, которые показывают расположение ячеек при отсутствии границ таблицы. На печать линии сетки не выводятся, они нужны только для облегчения редактирования таблицы.

Для обрамления таблицы необходимо воспользоваться командой **Таблица/Свойства таблицы**. В появившемся окне выбрать опцию *Обрамление*. В разделе положение линий можно задать произвольное обрамление ячеек или таблицы целиком. В разделе окна линия – стиль можно выбрать сплошную линию, штриховую, двойную и т.п., а также выбрать ее толщину.

### Задание 2

Создать таблицу: столбцов – 2, ячеек – 4

Введите в ячейки данные:

Наименование	объем продаж
Выпечка	105
Мороженое	232
Конфеты	211

Добавьте в таблицу новые строки (в конец) и введите данные

Лимонад	540
Соки	172
Печенье	234

Добавьте столбец в конце таблицы и введите данные:

Стоимость: 525, 2320, 21100, 10800, 688, 2560

Поставьте строку «лимонад» в конец таблицы

Поменяйте местами столбцы «Объем продаж» и «Стоимость»

Скопируйте строку «мороженое» в конец таблицы

Скопируйте таблицу 3 раза, начиная каждый раз с новой страницы

Оформите каждую таблицу, используя новый стиль

Отформатируйте первую таблицу:

Установите размер столбцов 7 – 4 – 6

Текст первой строки расположить по центру

### **Лабораторная работа №16 - Турбо паскаль. экран интегрированной среды**

*Цель:* изучить Турбо Паскаль. Ознакомиться с экраном интегрированной среды

*План:*

- 1) Структура экрана интегрированной среды.
- 2) Работа в окне редактирования
- 3) Вход в интегрированную среду.
- 4) Работа с блоками при редактировании текста.
- 5) Поиск с целью замены

#### *Методические рекомендации*

Осуществляется посредством активизации файла br.exe, который, как правило, записан в каталоге C:\BP\BIN (или D:\BP\BIN)

Структура экрана интегрированной среды:

основное меню – верхняя строка. Активизируется клавишей <F10> или щелчком мыши. С помощью клавиш управления курсором (<<=>, <↑>, <=>, <↓>, <Home>, <End>) выбирается нужный пункт меню и раскрывается клавишей <Enter>. В раскрытом "падающем" меню выбирается нужный пункт и запускается также клавишей <Enter>. Можно заметить, что напротив наименований некоторых пунктов "падающего" меню записаны

функциональные клавиши. Они позволяют выполнять нужные действия, не используя меню, что значительно повышает быстродействие при работе;

поле для размещения открываемых окон, в которых могут размещаться любые файлы, например с программами, исходными данными или результатами;

строка состояния – нижняя строка. Отражает состояние вычислительного процесса, представляет подсказку по назначению функциональных клавиш.

Выход из интегрированной среды. Осуществляется с помощью меню File/Exit. Это означает, что необходимо в основном меню выбрать пункт File, войти в него с помощью клавиши <Enter>. Затем в "падающем" меню выбрать пункт Exit и запустить его на выполнение той же клавишей <Enter>. Можно выйти из интегрированной среды и без помощи меню, нажав клавиши <Alt-X>.

В дальнейшем аналогичные действия не будут описываться столь подробно, а будут представляться в виде:

File/Exit или <Alt-X>

Работа в окне редактирования.

Открытие файла в окне редактирования. Осуществляется через

File/Open/ имя файла или <F3>

При этом на экране открывается содержимое файла и указывается номер окна.

*Замечания:*

1) "имя файла" можно записать в строке «Name» или выбрать его из предлагаемого ниже списка, куда курсор перемещается клавишей <Tab> или <Enter>. Выбранный файл открывается клавишей <Enter>;

2) в предлагаемом списке выводятся файлы с расширением .pas. При желании можно вызвать любые файлы, указав соответствующий формат группы файлов в строке «Name».

Создание нового файла. Осуществляется через File/New.

В результате открывается окно с именем "Noname" – "без имени". В этом окне можно записать текст файла, но для его сохранения ему нужно дать имя.

Задание имени файлу (или переименование файла) осуществляется через

File/Save as/ имя файла

*Замечания:*

1) если создается программа, в "имени файла" расширение можно не указывать. При этом автоматически файлу присваивается расширение .pas. Если файлу нужно другое расширение, то его необходимо указать в "имени файла";

2) при переименовании файла возможна его пересылка в другой каталог или диск, например на дискету. Для этого в "имени файла" записывается полное имя файла с указанием пути к нему, например: A:\MY\primer.pas (в этом случае будет создан файл primer.pas в каталоге MY на диске A:).

Сохранение файла. Осуществляется по окончании работы с файлом и периодически во время работы с ним для предупреждения неприятностей, связанных с непредвиденным отключением электроэнергии, в результате чего вся информация в файле, записанная после последнего сохранения может быть утеряна.

Сохранение файла осуществляется через

File/Save или <F2>

Работа с блоками при редактировании текста.

Для того, чтобы скопировать, переместить или удалить часть текста его необходимо сначала выделить.

Осуществляется клавишами перемещения курсора при нажатой клавише <Shift>:

<Shift- ←>, < Shift- ↑>, < Shift- →>, < Shift- ↓>, < Shift- Home>, <Shift- End>.

Снятие выделения осуществляется аналогично.

Для выделения блока текста с помощью мыши необходимо подвести указатель мыши к началу области выделения и, удерживая нажатой левую кнопку мыши, перетащить указатель мыши по выбранному фрагменту текста.

Для снятия выделения необходимо щелкнуть мышью по любому невыделенному участку текста.

Удаление фрагмента без восстановления. Фрагмент выделить и нажать

Edit/Clear или <Ctrl - Delete>

Удаление фрагмента в буфер. Фрагмент выделить и нажать

Edit/Cut или <Shift - Delete>

Копирование фрагмента в буфер. Фрагмент выделить и нажать

Edit/Сору или <Ctrl - Insert>

Восстановление фрагмента из буфера. Курсор поставить на место восстановления фрагмента и нажать

Edit/Paste или <Shift - Insert>

Для того чтобы скопировать фрагмент текста программы его необходимо сначала скопировать в буфер, затем установить курсор в нужное место программы и восстановить фрагмент из буфера. Для переноса фрагмента текста программы сначала необходимо удалить фрагмент в буфер, а затем восстановить фрагмент из буфера в нужном месте программы.

*Замечания:*

- восстановление фрагмента из буфера возможно как после удаления фрагмента в буфер, так и после его копирования в буфер;
- удаленный фрагмент восстановлению не подлежит;
- в буфере хранится только последний внесенный туда фрагмент;
- для восстановления любого фрагмента, внесенного ранее в буфер можно воспользоваться окном содержимого буфера, для чего следует нажать

Edit/Show clipboard

С содержимым окна буфера можно работать как с любым текстом, то есть можно редактировать, вставлять в текст программы и т.д.

Поиск с целью замены.

Этот режим используется в случае необходимости многократной замены одного набора символов на другой в тексте файла. Для этого необходимо:

- 1) войти в меню Search/Replаce;
- 2) ввести в первую строку слово или сочетание символов для поиска;
- 3) с помощью клавиши <Tab> перевести курсор во вторую строку и ввести в нее новое сочетание символов, которым следует заменить старое;
- 4) с помощью клавиши <Tab> перевести курсор на Change all, что произведет глобальную указанную замену во всех вхождениях искомого набора символов от курсора до конца текста. При этом предлагается перед каждой заменой подтвердить согласие на нее клавишей <Enter> или отказ от нее.

*Замечания:*

Поиск и замена будет осуществляться, начиная с позиции курсора и до конца текста программы.

Работа с окнами. Переход в заданное окно:

<Alt - № окна>

Закрытие окна редактирования:

Window/Close или <Alt - F3> или пиктограмма для мыши [■]

Открытие пользовательского экрана (на этот экран обычно выводятся результаты вычислений программы): <Alt - F5>. Для возвращения в окно редактирования можно нажать любую клавишу.

Размещение окон встык (используется при работе с несколькими файлами):

Window/Tile.

Размещение окон с перекрытием друг друга:

Window/Cascade.

Удаление (закрытие) всех окон:

Window/Close all.

Изменение размеров окна:

Window /Size/Move/<Shift - ←, ↑, →, ↓> или <Ctrl - F5>

Для фиксации выбранного размера служит клавиша <Enter>.

Перемещение окна:

« Window »/<Size/Move>/<←, ↑, →, ↓> или <Ctrl - F5>

Для фиксации выбранного положения окна служит клавиша <Enter>.

Распахивание окна во весь экран:

Window/Zoom или <F5>

Активизация следующего окна:

Window/Next или <F6>

Активизация предыдущего окна:

Window/Previous или <Shift - F6>

Открытие списка окон:

Window/List или <Alt - 0>

Запуск программы на выполнение.

Run/Run или <Ctrl - F9>

После запуска программы на выполнение результаты работы программы выводятся на экран пользователя <Alt - F5>. Для возвращения в окно редактирования можно нажать любую клавишу.

### ЗАДАНИЕ

1. Войти в интегрированную среду Турбо Паскаль.
2. Создать файл с именем XYZ781-1,  
где символы X, Y, Z представляют Ваши собственные инициалы в латинском алфавите и их нужно заменить на соответствующие буквы;  
781 - означает номер группы (ввести свой номер группы);  
1 - означает порядковый номер созданного Вами файла.
3. В содержимое файла записать программу:  
PROGRAM integr\_sreda;  
BEGIN  
    writeln (' Это моя первая программа');  
    writeln (' Тема: "Интегрированная среда" ');  
    writeln (' Я студент первого курса АмГУ ');  
END.
4. Сохранить созданную программу.
5. Запустить программу на выполнение.
6. Посмотреть на экране и записать в тетрадь результат выполнения программы.
7. Средствами работы с блоком скопировать предпоследнюю строчку программы 4 раза перед словом END.
8. Средствами глобальной замены в последних трех скопированных строчках заменить "АмГУ" на "МГУ".
9. Запустить программу. Посмотреть результат ее выполнения.
10. Средствами работы с блоком перенести строчку "writeln (' Это моя первая программа');" в положение перед словом END.
11. Средствами работы с блоком удалить строчку "writeln (' Тема: "Интегрированная среда" ');" в буфер.
12. Открыть окно содержимого буфера и скопировать строку " ' Тема: "Интегрированная среда" " из этого окна в текст предпоследней строки программы в любое место между ( ).

14. Расположить окна с программой и содержимым буфера встык.
15. Вернуться к каскадному расположению окон.
16. Изменить размеры и расположение окон для представления их на экране в виде:



17. Распахнуть окно с программой во весь экран.
18. Вернуться к прежнему расположению окон. Проверить различные варианты перемещения между окнами. Сохранить программу.

#### **Самостоятельная работа**

1. Изучить меню интегрированной среды Турбо Паскаль.
2. Ознакомиться с приемами редактирования текста программы.
3. Самостоятельно создать программу вывода на экран текста. Сохранить ее и запустить на выполнение.
4. Расположить окно с созданной программой различным образом, открыть данную программу в нескольких окнах.

#### **Лабораторная работа № 17 - Структура программы. компиляция программы**

Цель: научиться оформлять текст программы

План:

- 1) Оформление текста программы.
- 2) Перемещение курсора по экрану.
- 3) Компиляция программы.

#### *Методические рекомендации*

Оформление текста программы.

компилятор воспринимает строки до 126 символов;

клавиша <Enter> переводит курсор на новую строку в положение начала предыдущей строки;

группа операторов, выполняющих законченное действие, выделяется пустыми строками до и после нее;

идентификаторы могут включать строчные и заглавные буквы английского алфавита, цифры, знак подчеркивания "\_";

для семантически связанных групп операторов используются комментарии в фигурных скобках {}.

не следует давать локальным и глобальным переменным одинаковые имена.

Перемещение курсора по экрану. на одно слово вправо / влево: <Ctrl - →>, <Ctrl - ←>;

на один экран вверх / вниз: <Page Up>, <Page Down>;

скроллинг на одну строку вверх / вниз: <Ctrl - W>, <Ctrl - Z>;

к началу / концу строки: <Home>, <End>;

на первую / последнюю строку окна: <Ctrl - Home>, <Ctrl - End>;

к началу / концу текста: <Ctrl - Page Up>, <Ctrl - Page Down>.

Компиляция программы.

Компиляция предусматривает:

проверку программы на правильный синтаксис с указанием места и типа ошибки;

преобразование файла с языка высокого уровня в машинно-ориентированный, доступный для прочтения и выполнения машиной; если скомпилированный файл образует на диске, то его расширение **.pas** заменяется на **.exe**.

Местоположение скомпилированного файла. Зависит от установка в меню Compile/Destination. Если в указанном пункте установлено «Destination Memory», то скомпилированный файл (файл с расширением .exe) будет храниться только в памяти компьютера. Если установлено «Destination Disk», то файл можно найти в одном из каталогов диска.

*Замечание:* Если в меню Compile нет опции Destination, то скомпилированный файл (файл с расширением .exe) автоматически образуется в одном из каталогов диска.

Местоположение скомпилированного файла устанавливается в первой строке открытого меню Options/Directories.

Для сохранения введенной установки местоположения скомпилированных файлов используется меню Options/ Save BP.tp.

Компиляция программы осуществляется:

- при запуске программы на выполнение;
- через меню Compile/Compile или <Alt - F9>.

Действия для получения скомпилированной программы в виде .exe - файла:

- задать расположение .exe - файла на диске (если такая установка есть);
- установить местоположение .exe - файла, войдя в меню Options/ Directories

(см. п. 3.2.);

скомпилировать файл (см. п. 3.3.);

выйти из среды Турбо Паскаль и найти в установленном каталоге .exe – файл;

запустить .exe – файл на выполнение из операционной системы MS DOS или WINDOWS (нажав клавишу <Enter> на .exe – файле).

*Замечание:* .exe - файл нужен для возможности его запуска из операционной системы MS DOS или WINDOWS, не заходя в интегрированную среду Турбо Паскаль.

### ЗАДАНИЕ

1. Изучить пример программы вычисления значения переменной Z

$$Z = a \cdot x^2 + \frac{b}{\sqrt{\pi \cdot y}} - 18.01$$

при произвольно заданных в процессе выполнения программы значениях переменных X и Y и заданных значениях констант: a = 100, b = 3,2.

Комментарии по программе	Пример программы
Заголовок программы. Имя должно быть лат.буквами	PROGRAM struc_program;
Раздел констант	CONST a=100; b=3.2;
Раздел переменных. X и Z заданы вещественными, Y – целое	VAR x, z:real; y: integer;
Начало раздела операторов	BEGIN {Начало работы программы}
Операторный ввод данных в диалоговом режиме	write (' Введите любое значение X = '); readln (x); write (' Введите целое значение Y = '); readln (y);
Вычисления	z:=a * sqr(x)+b/sqrt(y * PI) —18.01;
Вывод результатов: в экспоненциальной форме в формате 5:2 (5 знаков, из них 2 – после запятой)	writeln (' Результаты вычислений '); writeln ('Z=' ,Z); writeln ('Z=' ,Z:5:2);
без знаков после запятой в экспоненциальной форме	writeln ('Z=' ,Z:5:0); writeln ('Z=' ,Z:3);
Конец раздела операторов	END. {Конец работы программы}

2. Записать программу в файл XYZ781-3. Запустить программу и получить результаты. Проверить работу программы при различных исходных данных.

### Лабораторная работа № 18 - Операторный ввод - вывод данных

*Цель:* разработка программы по индивидуальному заданию

*План:*

1. Разработать программу по индивидуальному заданию записать ее в файл XY0-4.

Вариант 1

Определить пылепроницаемость швов ткани  $\Pi_{ш}$ , г/м<sup>2</sup>:

$$\Pi_{ш} = \Pi - \Pi_{т},$$

где  $\Pi_{т}$  – пылепроницаемость ткани г/м<sup>2</sup>;

$\Pi$  – пылепроницаемость ткани со швом, г/м<sup>2</sup>.

$$\Pi_{т} = (M_0 - M) / S; \quad \Pi = (M_0' - M') / S',$$

где  $M_0$  – масса пробы ткани до испытания.  $M_0 = 6,6$  г;

$M$  – масса пробы ткани после наполнения пылью.  $M = 7,5$  г;

$M_0'$  – масса пробы ткани со швом до испытания.  $M_0' = 6,9$  г;

$M'$  – масса пробы со швом после наполнения пылью.  $M' = 8,01$  г;

$S, S'$  – площади проб, соответственно до и после испытания, м<sup>2</sup>:

$$S = S' = a \cdot b \cdot 10^{-6}$$

где  $a, b$  – размеры образцов:  $a = 126$  мм,  $b = 176$  мм.

*Методические рекомендации*

В программе должны быть учтены следующие условия:

- ввод данных осуществить с помощью операторов в диалоговом режиме;
- вывод результатов должен быть представлен в удобном для прочтения виде с комментариями о наименовании и единицах измерения получаемых величин;
- в качестве результатов должны быть выведены не менее трех переменных;
- в выводе результатов указать, по каким исходным данным получены результаты.

Получить **.exe** - файл. Найти его местоположение на диске. Убедиться в работоспособности программы из операционной системы MS DOS или WINDOWS.

*Самостоятельная работа*

1. В тетрадь записать:

- программу;
- комментарии по программе аналогично примеру в лабораторной работе № 22;
- результаты выполнения программы;
- местоположение **.exe** - файла;
- объем памяти, занимаемой файлами с расширением **.exe** и **.pas**.

2. Файл программы записать на дискету.

3. Самостоятельно внести изменения в задание и отразить эти изменения в созданной программе.

### Лабораторная работа № 19 - Файловый ввод данных

*Цель:* освоить файловый ввод данных

*План:*

Работа выполняется по индивидуальному заданию

*Методические рекомендации*

Считывание данных из файла обеспечивается включением в программу следующих элементов.

В разделе переменных следует объявить две переменные:

- файловую типа **text** для организации посредством нее связи между программой и файлом исходных данных;
- типа **string** для хранения имени файла исходных данных. Данная переменная может отсутствовать, если в тексте программы указано конкретное имя файла исходных данных.

Пример:

```
VAR n1: string; f1:text;
```

В разделе операторов следует организовать:

- ввод имени файла в диалоговом режиме и запись его в переменную типа **string**.

Пример:

```
write (' Введите имя файла исходных данных');
```

```
readln (n1);
```

На возникающий в процессе выполнения программы запрос необходимо ввести имя файла исходных данных, например **ttt.dat**.

*Замечание:* если программа работает только с одним файлом, то данный запрос не нужен.

- связь между файловой переменной и файлом исходных данных с помощью процедуры **assign (f1,n1)**.

*Замечание:* если программа работает только с одним файлом, то вместо переменной **n1** можно записать конкретное имя файла, например **assign (f1, ttt.dat)**.

- открытие файла для считывания из него данных с помощью процедуры **reset(f1)**;
- считывание данных из файла с помощью оператора ввода данных, например **readln(f1,a,b,c)**. Данный оператор считывает из файла с помощью файловой переменной **f1** значения переменных **a,b,c**, поэтому он должен стоять до первого использования считываемых переменных.

*Замечания:*

1) на первом месте в операторе **readln** ставится имя файловой переменной, на втором – перечисляются идентификаторы тех переменных, чьи значения нужно считать из файла;

2) считываемые из файла переменные в операторе **readln** перечисляются в том порядке, в каком записаны их значения в файле исходных данных.

3) если исходные данные считываются из файла, то их ввод в диалоговом режиме с клавиатуры не нужен;

4) вид оператора **readln** зависит от расположения данных в файле исходных данных. Если данные записаны в одну строку, то их считывание возможно одним оператором **readln**, если в несколько строк – то и операторов должно быть несколько.

5) вид файла исходных данных для оператора **readln(f1,a,b,c)**:

```
23 2 1.3
```

При этом после выполнения оператора **readln** в переменную **a** запишется значение 23, в переменную **b** – 2, в переменную **c** – 1.3.

- закрытие файла исходных данных с помощью процедуры **close (f1)**, которую можно поставить в программе после считывания всех данных из файла или, как чаще всего и делают, перед закрытием программы словом **END**.

*Самостоятельная работа*

Отредактировать программу из лабораторной работы № 23, заменив ввод данных в диалоговом режиме с клавиатуры на ввод из файла. Файл исходных данных назвать **date\_4.dat**.

## **Лабораторная работа № 20 - Вывод результатов в различных вариантах**

*Цель:* используя лекционный материал, самостоятельно осуществить вывод результатов в различных вариантах

*Методические рекомендации*

Для вывода результатов используются операторы **write()**, **writeln()**, где в () указываются данные с различными форматами вывода.

**ЗАДАНИЕ**

Отредактировать программу из лабораторной работы №24, организовав вывод результатов в программе в следующих вариантах:



1. число  
число  
число
2. число число число
3.  $x = \text{число}$   $y = \text{число}$   $z = \text{число}$
4.  $x = \text{число}^{***}$   $y = \text{число}^{***}$   $z = \text{число}$
5.  $x = \text{число}$   
\*\*\*\*\*  
 $y = \text{число}$   
\*\*\*\*\*  
 $z = \text{число}$
6.  $x = \text{число, ед.изм.}$ ,  $y = \text{число, ед.изм.}$ ,  $z = \text{число, ед.изм.}$
7. Наименование  $x = \text{число, ед.изм.}$ ,  
Наименование  $y = \text{число, ед.изм.}$ ,  
Наименование  $z = \text{число, ед.изм.}$

Наименование	Единицы измерения	Обозначение	Значение
Скорость	м/с	x	3.05

9. В формате с точностью до 3-х знаков после запятой:

$x = \_ \text{число}$ ,  $y = \_ \text{число}$ ,  $z = \_ \text{число}$

10. В экспоненциальной форме

$x = \text{число}$   $y = \text{число}$   $z = \text{число}$

11. число

число

число

12. число \_ \_ \_ \_ \_ число

число

Вывести результаты и исходные данные с комментарием в столбик с точностью до 5-го знака после запятой.

Добавить в таблицу вывод исходных данных.

Самостоятельно разработать несколько вариантов вывода.

### Лабораторная работа № 23 - Стандартные функции

*Цель:* научиться работать со стандартными функциями

*Задание:*

- 1) Найти значение функции  $z(t)$  при заданном значении  $t$ . Для этого записать выражение  $z(t)$ , используя стандартные функции Турбо Паскаля.
- 2) Определить значения указанных в задании стандартных функций, обозначив их  $z1$ ,  $z2$ ,  $z3$ ,  $z4$ .
- 3) Определить принадлежит ли заштрихованной области точка с координатами  $(x, y)$ . Для этого записать выражение, определяющее принадлежность точки к заштрихованной области.
- 4) Результаты вывести на экран в виде:

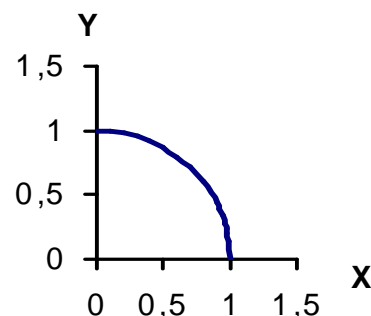
При  $t = -1.57$   $z = -15.547$

$z1 = -15$   $z2 = -16$   $z3 = FALSE$   $z4 = FALSE$

Точка  $(0.5; 1.2)$  лежит в заштрихованной области – *FALSE*.

Пример задания

1.  $z(t) = 3^{-t+1} \cdot \sin t$  при  $t = -1,57$
2.  $z1$  – целая часть от числа  $z$ , полученная путем отбрасывания дробной части.  
 $z2$  - целое значение, полученное из числа  $z$  путем его округления  
 $z3$  - определяет положительно ли  $z$ .  
 $z4$  – проверяет нечетность числа  $z2$ .
3. (0,5;1,2)



Программа по индивидуальному заданию:

```
PROGRAM PRIMER;
VAR  x, y, z, t: real; z1, z2: integer; z3, z4: boolean;
BEGIN
  write ('Введите t=');           {Ввод в диалоговом режиме значения
  readln (t);                     переменной t}
  write ('Введите координаты точки x='); readln (x);   {Ввод координат точки}
  write ('Введите координаты точки y='); readln (y);
  z :=exp((-t+1)*ln(3))*sin(t);    {Расчет значения функции z}
  writeln ('Программу составил Иванов И.И. ');
  writeln ('при t=', t:5:2, ' z=', z:6:3);    {Вывод значения функции z при
заданном значении t}
  z1:=trunc(z); z2:=round(z);      {Расчет значений z1, z2, z3, z4}
  z3:=z>0; z4:=odd(z2);
  writeln ('z1=', z1:4, ' z2=', z2:4, ' z3=', z3, ' z4=',z4); {Вывод значений z1, z2,
z3, z4}
  write ('Точка ('x:3:1;', 'y:3:1,') лежит в заштрихованной области- ');
  writeln ((x>0) and (y>0) and (sqr (x)+sqr(y)<=1));
END.
```

**Выражение различных функций через стандартные функции языка Паскаль**

Функция	Функция на языке Паскаль
$y=x^z$	$y:=\exp(z*\ln(x))$
$y=\arcsin x = \arctg \frac{x}{\sqrt{1-x^2}}$	$y:=\arctan(x / \text{sqr}(1-\text{sqr}(x)))$
$y = \arccos \quad x = \frac{\pi}{2} - \arctg \frac{x}{\sqrt{1-x^2}}$	$y:=PI / 2 - \arctan( x / \text{sqr}(1-\text{sqr}(x)))$
$y = \text{arcctg}x = \frac{\pi}{2} - \text{arctg}x$	$y:=PI / 2 - \arctan(x)$
$y = \text{sc}x = \frac{1}{\cos x}$ (секанс)	$y:=1 / \cos(x)$
$y = \text{cs}x = \frac{1}{\sin x}$ (косеканс)	$y:=1 / \sin(x)$

### Определение принадлежности точки указанной области

Для определения принадлежности точки указанной области необходимо:

- 1) определить уравнение границ области;
- 2) составить неравенства, определяющие заштрихованную область.

#### Пример 1

Уравнение прямой:

$$Y = ax + b$$

Для определения коэффициентов  $a$  и  $b$ , рассмотрим точки пересечения прямой с осями  $X$  и  $Y$ .

3. Точка  $B(0;2)$

Подставим координаты т.В в уравнение прямой и определим значение коэффициента  $b$ :

$$x=0, y=2$$

$$2 = a \cdot 0 + b$$

$$b = 2$$

4. Точка  $A(2;0)$

Подставим координаты т.А и значение коэффициента  $b=2$  в уравнение прямой и определим значение коэффициента  $a$ :

$$0 = a \cdot 2 + 2$$

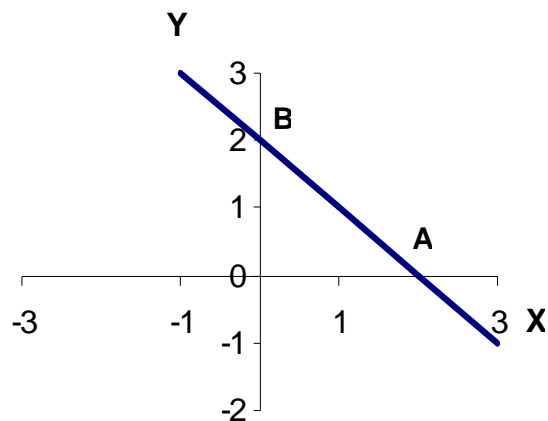
$$a = -1$$

Следовательно, уравнение прямой будет иметь следующий вид:

$$y = -x + 2.$$

Для всех точек, находящихся снизу от прямой, будет выполняться следующее неравенство, определяющее условие заштрихованной области:

$$y \leq -x + 2.$$



#### Пример 2

Уравнение границ:

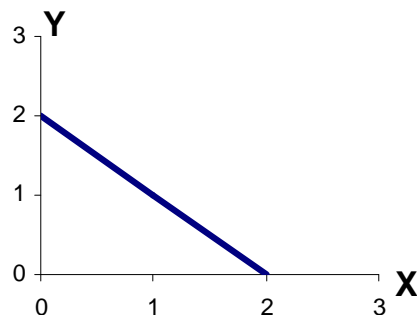
$$y = -x + 2; x = 0; y = 0$$

Условия заштрихованной области:

$$y \leq -x + 2;$$

$$x \geq 0;$$

$$y \geq 0$$



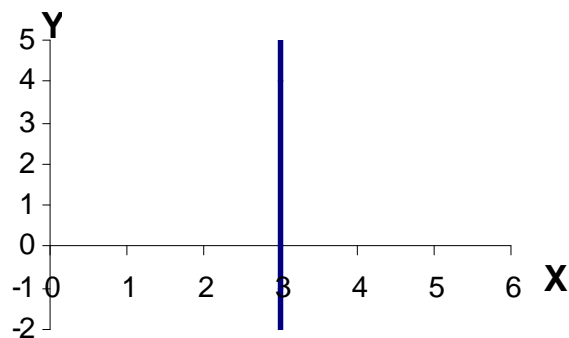
#### Пример 3

Уравнение границы:

$$x = 3$$

Условие заштрихованной области:

$$x \geq 3$$



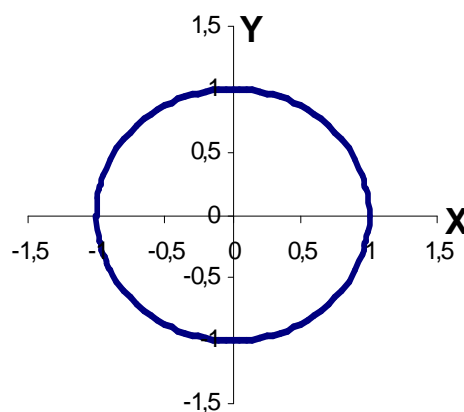
Пример 5

Уравнение границ:

$$x^2 + y^2 = 1$$

Условие заштрихованной области:

$$x^2 + y^2 \leq 1$$



Пример 6

Уравнение границ:

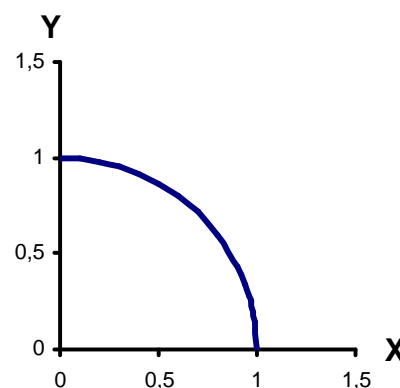
$$x^2 + y^2 = 1; x=0; y=0$$

Условие заштрихованной области:

$$x^2 + y^2 \leq 1;$$

$$x \geq 0;$$

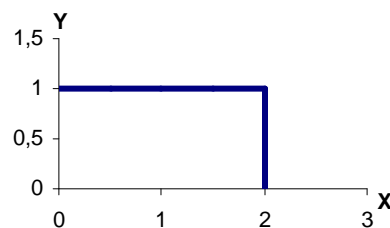
$$y \geq 0$$



## ЗАДАНИЕ

### Вариант 1

- $z = 2^{-t} \cdot \sqrt{t + 4\sqrt{|t|}}$  при  $t = 4,741$
- $z1$  – соответствует модулю  $z$ ;  
 $z2$  – число, полученное при округлении  $z$ ;  
 $z3$  – определяет четное ли значение  $z2$ ;  
 $z4$  – символ, стоящий под кодом 136.
- $(0,5;0,5)$



### Самостоятельная работа

- Использую таблицу стандартных функций, рассчитать любые значения переменных  $z5, z6, z7$ .
- Изменить границы заштрихованной области и координаты точки. Определить принадлежность точки данной заштрихованной области.
- Вывод результатов осуществить в новый файл результатов.

## Лабораторная работа № 24 - Турбо пascal. Работа с массивами. Оператор цикла for

Цель: Разработать программу по индивидуальному заданию

Задание

. В программе использовать:

- Описание массива любым способом (через раздел констант, через раздел типов или через раздел переменных).
- Файловый ввод данных. При этом файл исходных данных должен иметь вид: N

a1 a2 a3 ... ak

где N – число элементов массива

a1, ..., ak – значения элементов массива.

3. Считывание данных из файла исходных данных с помощью оператора цикла, например:

```
for i:=1 to N do read (f1, A[i]);
```

где i – переменная, которая считает элементы массива и организует цикл;

f1 – файловая переменная, связанная с файлом исходных данных;

A[i] – элементы исходного массива.

4. Преобразование элементов исходного массива A[i] в элементы нового массива B[i], например:

```
B[i]:=A[i]*5;
```

где каждый элемент нового массива B[i] больше соответствующего элемента исходного массива A[i] в 5 раз.

#### Замечание

Любые действия над элементами массива рекомендуется выполнять в рамках оператора цикла:

```
for i:=1 to N do begin ... end;
```

5. Файловый вывод данных.

Пример файла результатов:

Исходный массив:

1 2 3 4 5

Массив результатов 1:

25 10 15 20 25

Массив результатов 2:

0.1 0.2 0.3 0.4 0.5

#### *Методические указания*

увеличение элементов исходного массива A в 5 раз с использованием файлового ввода данных.

Файл исходных данных GGG.TXT имеет вид:

5

1 2 3 4 5

```
PROGRAM massiv;{ Увеличение элементов массива в 5 раз }
```

```
VAR a,d:array[1..5]of integer; N,i:integer; f1:text;
```

```
BEGIN
```

```
  assign (f1,'GGG.TXT');
```

```
  reset(f1);
```

```
  readln (f1,N); {Считывание из файла исходных данных количества  
                 элементов массива }
```

```
  writeln;
```

```
  writeln('Исходный массив:');
```

```
  for i:=1 to N do
```

```
    begin
```

```
      read(f1,A[i]); {Считывание значений каждого  
                   элемента исходного массива }
```

```
      write(A[i]:4); {Вывод на экран каждого элемента  
                   исходного массива }
```

```
    end;
```

```
  writeln;
```

```
  writeln('Массив результатов 1:');
```

```
  for i:=1 to N do
```

```
    begin
```

```
D[i]:=A[i]*5; {Вычисление значений каждого
                элемента массива результатов}
write(D[i]:4); {Вывод на экран каждого элемента
                массива результатов}
```

end;

END.

#### Самостоятельная работа

1. Найти массив результатов 3 самостоятельно любым способом.
2. Вывести все полученные массивы в новый файл результатов.
3. Расположить полученные массивы в файле в строчку, в столбик, используя таблицу.

### Лабораторная работа № 25 - Турбо паскаль: работа с двумерными массивами

Цель: особенности работы с двумерными массивами в Турбо Паскале

#### Методические рекомендации

Точки A(x1, y1), B(x2, y2) и C(x3, y3) перемещаются, изменяя свои координаты. Известны их координаты в 8 моментах времени (см.карточку индивидуального задания).

1. Найти расстояние АВ, АС, ВС в каждый момент времени и вывести его на экран в виде таблицы.

Момент времени	АВ	АС	ВС
1			
2			
...			
7			
8			

Расстояние между точками A(x1, y1) и B(x2, y2) определяется по формуле:

$$AB = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (1)$$

С учетом того, что расстояния АВ, АС и ВС в данном случае будут представлять одномерный массив, то их можно определить:

```
AB[i]:=sqrt(sqrt(x1[i,j]-x2[i,j])+sqrt(y1[i,j]-y2[i,j]));
```

2. Ввод элементов исходного массива осуществлять из файла.

Считывание элементов из файла и вывод результатов на экран осуществлять оператором цикла.

Пример организации считывания данных из файла ttt.txt вида:

```
A11 A12 A13 A14
```

```
A21 A22 A23 A24
```

где Aij – значения элементов массива;

При этом первый столбец можно считать координатой x1, второй – y1, и т.д.

i – номер строки массива;

j – номер столбца массива.

С учетом выбранной формы файла исходных данных формула (1) запишется в виде:

```
AB[i]:=sqrt(sqrt(A[i,1]-A[i,3])+sqrt(A[i,2]-A[i,4]));
```

Фрагмент программы:

```
VAR A:array(1..2,1..3)of real;
```

```
f1:text; i,j:integer;
```

```
BEGIN
```

```
assign(f1,'ttt.txt'); reset(f1);
```

```
for i:=1 to 2 do
```

```

begin
  for j:=1 to 3 do read(f1,A[i,j]);
  readln(f1);
end;

```

.....

```

close(f1);
END.

```

3. После отладки программы организовать вывод результатов в файл.

#### *Самостоятельная работа*

1. В файл исходных данных добавить произвольные координаты точки D.
2. Рассчитать расстояния AD, BD, CD в каждый момент времени, вывести его в таблицу.
3. Создать новый файл результатов, в котором расположить таблицу результатов в горизонтальном виде.

### **Лабораторная работа №26 - Турбо паскаль: условные операторы**

*Цель:* получить навыки работы с условными операторами

*Задание*

- 1) Составить блок-схему работы программы для выполнения следующего задания:
- 2) Создать программу расчета по предложенному заданию с организацией ввода исходных данных с клавиатуры и вывода результатов на экран.
- 3) Проверить работоспособность программы.

$$y = \begin{cases} 25 \ln 2x & x \leq 2 \\ x^3 + 2x, & \text{если } 2 < x < 8 \\ \sqrt{x} + x^2 & \text{в остальных случаях} \end{cases}$$

- 4) Отредактировать программу с организацией вывода результатов в файл. В файле результатов должны быть указаны исходные данные для расчета и комментарии к результатам расчета.
- 5) Оформить в тетради текст программы.
- 6) Записать в тетрадь имя файла результатов работы программы и его содержание.

#### **Методические рекомендации**

Дано значение  $x$ , рассчитать значение величины  $w$  в соответствии с заданием

$$w = \begin{cases} 0 & x \leq 0 \\ 2x & \text{если } 0 < x < 4 \\ x^2 & 4 \leq x < 10 \\ x & \text{в остальных случаях} \end{cases}$$

С целью правильного оформления условного оператора необходимо построить блок-схему, после чего разработать программу расчета.

В приведенном ниже блоке программы предлагается один из вариантов использования условного оператора:

```
BEGIN
```

```
...if x<=0 then w:=0
```

```
else if (x>0) and (x<4) then w:=2*x
```

```
else if (x>=4) and (x<10) then w:=sqr(x)
```

```
else w:=x;
```

#### **Самостоятельная работа**

Создать программу для индивидуального задания:

Вывести первое число, если оно больше второго, и оба числа, если это не так.

## Лабораторная работа № 27 - Турбо Паскаль: операторы цикла с постусловием и с предусловием

*Цель:* получить навыки работы с операторами с постусловием и с предусловием

*Задание:*

Вариант 1

1. Составить блок-схему для определения поверхностной плотности трикотажного полотна в зависимости от выбранного вида пряжи. Полотно выполнено переплетением интерлок из хлопчатобумажной или шерстяной пряжи линейной плотности 16,5 текс. Алгоритм определения поверхностной плотности,  $P_s$ , г/м<sup>2</sup>, приведен ниже.

$$l = \frac{\lambda \cdot \sqrt{T}}{31,62},$$

где  $\lambda = 23$  для хлопк  
 $\lambda = 24$  для шерсти

$$A = 0,13 \cdot l + 0,1 \sqrt{T}$$

$$B = 0,35 \cdot l - 0,09 \sqrt{T}$$

$$P_z = \frac{100}{A}$$

$$P_v = \frac{100}{B}$$

$$P_s = 10^{-4} \cdot P_z \cdot P_v \cdot l \cdot T$$

2. Записать условный оператор, реализующий вычисление по полученной блок-схеме.

3. Создать программу с организацией ввода исходных данных с клавиатуры и вывода результатов расчета на экран.

4. Проверить работоспособность программы.

5. Отладить программу с организацией расчета для пряжи линейной плотности в пределах от 16,5 до 29 текс с использованием оператора цикла с постусловием. Расчеты производить с шагом 1,5 текс.

6. Проверить работоспособность программы.

7. Отладить программу с организацией ввода исходных данных из файла и вывода результатов расчета в файл.

8. Выполнить расчеты по предложенному заданию.

9. Оформить отчет по работе в тетради с полным представлением окончательного варианта текста программы, указанием имени и содержания файлов исходных данных и результатов расчета.

### *Методические рекомендации*

Рассчитать запас материала на операции разгрузки, приемки и складирования. Расчет выполняется по следующей формуле:

$$z = \frac{K \cdot R \cdot D}{P} \cdot 100,$$

где  $K$  – коэффициент запаса,  $K = 0,025$

$R$  – однодневный расход,  $R = 800 \text{ м}^2$

$D$  – интервал поставок,  $D = 1 - 17$  дней, расчет производится с шагом 1 день.

$P$  – число рабочих, занятых данной работой,  $P = 1$

### *Фрагмент программы*

```
... d1:=1;d2:=17;h:=1
```

```
repeat
```

```
z:=k*R*d1/P*100;
```

```
Write('d=',d:4:2,' ');
```

```
Writeln('z=',z);
```



```
d1:=d1+h
until d1>d2;
```

...Самостоятельная работа

1. Создать программу для данного задания, используя оператор цикла с предусловием.
2. Вывести результаты работы программы в виде таблицы в новый файл результатов.

### Лабораторная работа №31 - Работа с электронной почтой и в телеконференциях

**Цель работы:** получить практические навыки по работе с электронной почтой и в телеконференциях. Научиться работать с почтовыми программами, уметь их настраивать.

**План:**

- 1) Научиться создавать электронный почтовый ящик
- 2) Научиться создавать и отправлять электронные письма
- 3) Научиться настраивать почтовую программу на компьютере.
- 4) Создание, отправка и получение сообщения с помощью почтовой программы.

**Должны знать:** правила пользования электронной почтой, этику общения в телеконференциях по темам; как создать почтовый электронный ящик на бесплатном российском сервере, как установить почтовую программу на примере MS Outlook.

**Должны уметь:** правильно работать с программой Outlook, выполнять основные установки, в том числе коммутацию с Интернет, создавать электронный почтовый ящик, настраивать его.

#### Методические указания.

За последнее время для работы с электронной почтой стали часто использоваться web-технологии. Появились web-сайты, которые предлагают всем желающим зарегистрировать свой бесплатный электронный почтовый ящик. Примером таких сайтов могут являться <http://mail.ru>, <http://www.rambler.ru>, <http://www.yandex.ru> Для работы с такой электронной почтой не требуются специальные почтовые программы, т.к можно воспользоваться любым браузером для загрузки отображения соответствующей web-страницы.

#### Задание №1. Научиться создавать электронный почтовый ящик.

1. Запустить на компьютере программу Internet Explorer. В навигационной строке ввести адрес любого российского сервера бесплатной электронной почты, например <http://www.yandex.ru>. (Рис.)

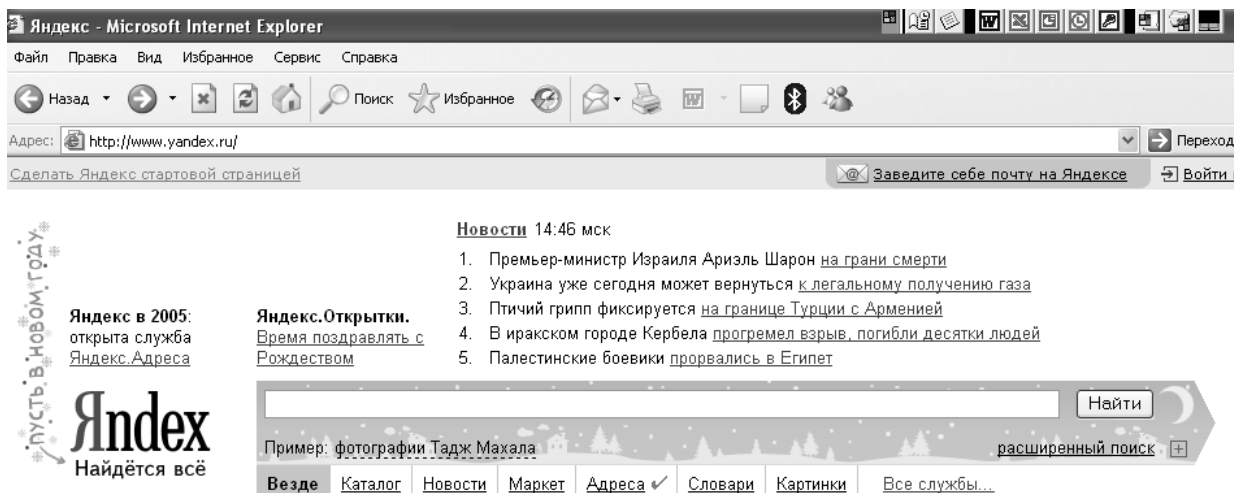


Рис. Бесплатный российский сервер электронной почты Яндекс.

2. Выбрать опцию «**Зарегистрироваться**». Для этого нажмите на гиперссылку «Заведи себе почту на Яндексе». В появившемся окне регистрации нового пользователя (Рис.31) необходимо создать имя почтового ящика. Оно состоит из двух частей – первая user\_name , имя пользователя, вторая – server\_name, имя почтового сервера, где происходит регистрация

Яндекс

Почта [Помощь](#)

[Персональные данные](#) [Платежные данные](#) [Настройка Яндекса](#)

### Регистрация: шаг 1 из 2

Потратив пару минут для регистрации на Яндексе, вы сможете получить неограниченный [почтовый ящик](#) без спама и вирусов, неограниченное пространство для [собственного сайта](#) и многие другие приятные возможности.

Логин:

должен состоять из символов A-z, 0-9, -, начинаться с буквы, заканчиваться буквой или цифрой и содержать не более 20 символов

Существует мнение, что все хорошие логины на Яндексе уже заняты. Но это не так. Мы можем посоветовать вам интересный незанятый логин. Для этого достаточно указать имя и фамилию (они понадобятся и для дальнейшей регистрации).

Ваше имя:

Фамилия:

Яндекс охраняет персональные сведения пользователей в соответствии с [Соглашением о конфиденциальности информации](#).

Рис.. Первый шаг регистрации почтового ящика.

3. Придумайте для своего почтового ящика **логин** – это имя пользователя электронной почты, первая часть почтового адреса. Логин имеет произвольный характер, главное, чтобы он был оригинален для того доменного имени почтового сервера, где происходит регистрация.

4. Продумайте пароль для входа в Ваш электронный почтовый ящик. Он не должен быть слишком простым, в то же время лично для себя его необходимо запомнить. Оптимальный вариант пароля – сочетание буквенных и цифровых символов, понятное только лично самому пользователю.

5. Если все-таки пароль будет забыт Вами с течением времени по какой-либо причине, то необходимо создать ответ на контрольный вопрос (Рис.32). Ответ на такой вопрос будете знать только Вы.

6. После заполнения всех необходимых полей формы регистрации выводится сообщение о том, что создан почтовый ящик для нового пользователя. При этом предлагается заполнить некоторые персональные данные (Рис.). Теперь можно войти в почту или написать первое письмо.

## Регистрация: шаг 2 из 2

Ваш логин - Len-iva444

Пароль:	<input type="password" value="••••"/>	<a href="#">Как правильно составлять пароль</a>
Пароль должен содержать не менее 4 символов из списка: A-z, 0-9, !@# \$ % ^ & * ( ) _ - + и не может совпадать с логином.		
Подтвердите пароль:	<input type="password" value="••••"/>	введено верно
Контрольный вопрос:	<input type="text" value="не выбран"/>	Если вы забудете пароль, вы сможете получить доступ, ответив на этот вопро
Ответ:	<input type="text" value="Девичья фамилия матери"/>	
Электронная почта:	<input type="text" value=""/>	Вы сможете использовать этот адрес при работе со службами Яндекса. Если вы введете адрес, на него будет выслан запрос о его подтверждении.
Контрольные цифры:	<input type="text" value="031"/>	Если вы не видите кнопку <ok> и картинку контрольными цифрами, это означает, что в вашем браузере отключена поддержка графики. Включите ее, перезагрузите страницу и заполните форму регистрации снова.


Нажимая кнопку <ok>, вы принимаете условия [Пользовательского соглашения!](#)

Рис. Второй шаг регистрации почтового ящика

Яндекс **почта** [Помощь](#) [Len-iva444](#) [Выход](#)

[Персональные данные](#) [Платежные данные](#) [Настройка Яндекса](#)

### Готово!

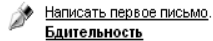


Регистрация успешно завершена.

Теперь у вас на Яндексе есть [свой паспорт](#). В нем хранятся ваши личные данные и предпочтения.

Вы можете начать пользоваться [сервисами Яндекса](#) прямо сейчас.

Просим вас найти минуту и рассказать немного о себе, чтобы Яндексу было легче вас обслуживать, а вам — удобнее пользоваться Яндексом.



Сотрудники Яндекса никогда и ни под каким предлогом не спрашивают пароль по телефону или электронной почте!

### Дополнительная информация

Обращение:	<input type="text" value="Не указано"/>
Псевдоним:	<input type="text" value=""/>
Пол:	<input type="radio"/> предпочитаю не указывать <input type="radio"/> мужской <input type="radio"/> женский
Ваш день рождения:	<input type="text" value="января"/>
Страна:	<input type="text" value="не указано"/>
Округ, город:	<input type="text" value=""/>

Рис. Результат создания электронного почтового ящика

## Задание №2. Научиться создавать и отправлять электронные письма.

1. Для проверки писем в почтовом ящике, а так же для того, чтобы написать письмо зарегистрированные пользователи должны ввести свой идентификатор (*логин*) и пароль, после чего они могут войти в почтовую систему.

2. Выберите опцию **Написать письмо** (Рис.34.). Заполните графы **Кому** и **Тема**. Графа **Кому** заполняется либо вручную, либо из адресной книги. Ссылка Адресная книга позволяет создать виртуальную записную книжку с основными данными всех нужных Вам адресатов. **Тема** письма так же должна быть заполнена. Незаполненное окно **Тема** говорит либо о некомпетентности пишущего адресата, либо о его халатности или пренебрежении. Поэтому «электронный этикет» требует обязательного заполнения Темы письма. Окно **От Кого** заполняется автоматически. Текст письма может быть любым, однако не помешает проверка грамматики и орфографии. (Кнопка «Проверить...»).

The screenshot shows the Yandex Mail interface for writing an email. At the top, the user is logged in as 'len-iva444'. The main heading is 'Написать письмо'. There is an 'Отправить' button with an envelope icon. The 'От кого:' field is set to 'Len-iva444@yandex.ru'. The 'Кому:' field contains 'alla-pti@narod.ru'. There are empty fields for 'Копия:' and 'Скрытая:'. The 'Тема:' field contains 'Ящик зарегистрировала!'. Below the fields, there are radio buttons for 'Простой текст (plain)' and 'С форматированием (html)'. The text area contains the following text: 'Здравствуйте, |', '---', 'Яндекс.Почта: объем почтового ящика не ограничен!', and 'http://mail.yandex.ru/monitoring/'. On the right side, there are links for 'Словарь Лингво' and 'Русская клавиатура'.

Рис.Пример написания электронного письма

3. Выберите Опцию (поставьте галочку) **Сохранить копию письма при отправке**. Существенной особенностью Web-почты является то, что все сообщения постоянно хранятся на удаленном сервере, а не на локальном компьютере пользователя. Это часто бывает удобно, т.к. не всегда есть возможность отправлять необходимые файлы со своего компьютера. Если копия письма сохранена в ящике, то вложенные в отправленные письма файлы могут быть доступны для Вас с любого компьютера, имеющего подключение к Интернет.

4. Добавьте в письмо вложенный файл. Для этого нажмите кнопку «Обзор» у некоторых серверов эта опция называется «Вложить» или «Прикрепить». Обзор позволяет найти нужный файл на компьютере (Рис). В данной работе вложите файл с отсканированным и распознанным документом (лабораторная работа №3) или файл с презентацией (лабораторная работа №4)

5. Отправьте готовое письмо с вложенным файлом. Это может занять некоторое время. Длительность отправки зависит от скорости линии, по которой имеется доступ в Интернет, а также от объема вложенного файла.

6. Дождитесь сообщения, что письмо отправлено (рис.36). Выберите опцию **Вернуться в текущую папку**.

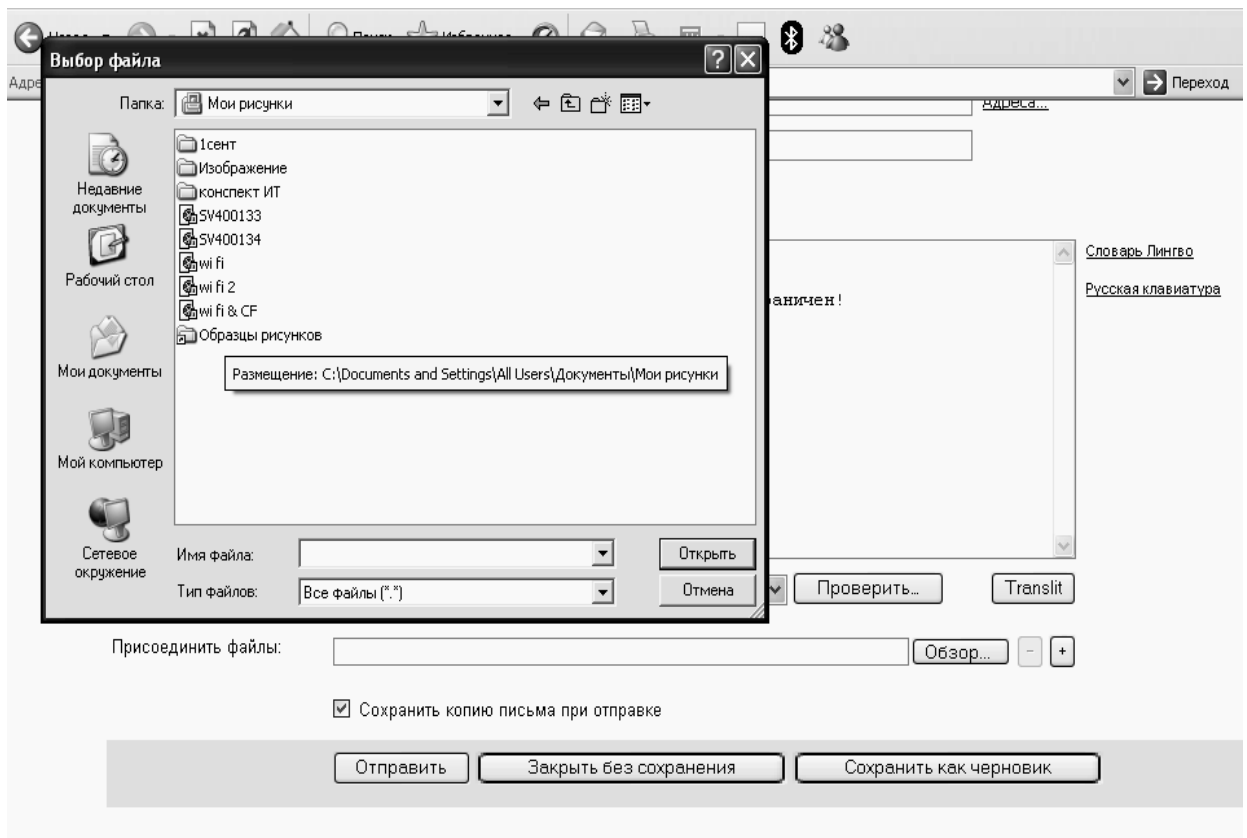


Рис. Пример написания электронного письма с прикрепленным (вложенным) файлом.



Рис. Интерфейс программы после успешного отправления письма.

7. Просмотрите имеющуюся почту. Все пришедшие на Ваш адрес письма находятся в папке **Входящие** (рис). Если Вы работали в паре со своим сокурсником, то письма Вы могли отправить друг другу. Таким образом, в Вашем почтовом ящике появилось новое письмо .



Рис. Электронная почта с Web-Интерфейсом. Папка Входящие

### Задание №3. Научиться настраивать почтовую программу на компьютере.

Для любых компьютерных платформ существует большое разнообразие почтовых программ. Среди них такие широко известные как Outlook Express, MS Outlook, The Bat! и другие. Суть работы такой программы в том, что адресат для получения письма должен соединиться с Интернетом и доставить почту со своего почтового ящика на удаленном почтовом сервере на свой локальный компьютер. При этом существует возможность доставки почты с нескольких ящиков, с нескольких почтовых серверов. Удобна также адресная книга и возможность одновременной рассылки писем нескольким адресатам.

В данной лабораторной работе предстоит настроить и использовать для работы с почтой программу MS Outlook. В первую очередь необходимо в соответствии с полученными в процессе регистрации почтового ящика данными (имя почтового ящика, пароль и др.) настроить почтовую программу. Создадим в почтовой программе Outlook Express учетную запись «Почта Интернета», при помощи которой можно будет отправлять и принимать электронную почту с созданного в первом задании почтового ящика.

1. Запустить программу MS Outlook. Появится окно, состоящее из 4 частей (Рис.38). В левой верхней части находится перечень папок с корреспонденцией и другие папки пользователя программы MS Outlook. Папки **Входящие**, **Исходящие**, **Отправленные**, **Удаленные** и **Черновики** аналогичны почтовой программе в web-интерфейсом. Пользователь может создавать собственные папки для хранения тематически сгруппированных сообщений. В папках могут храниться не только сообщения, но и файлы, созданные с помощью других приложений. В нижней левой части окна размещается список контактов, который предоставляет доступ к информации, хранящейся в **Адресной книге** (адреса электронной почты, телефоны и так далее).

Если программа не настроена и запускается впервые, то все папки из левого окна пустые.

MS Outlook, будучи программой не только для работы с электронной почтой, но и планировщиком задач, имеет еще **Календарь**, где пользователь разрабатывает планы и задачи, встречи, телефонные звонки на день или несколько дней вперед (Рис).

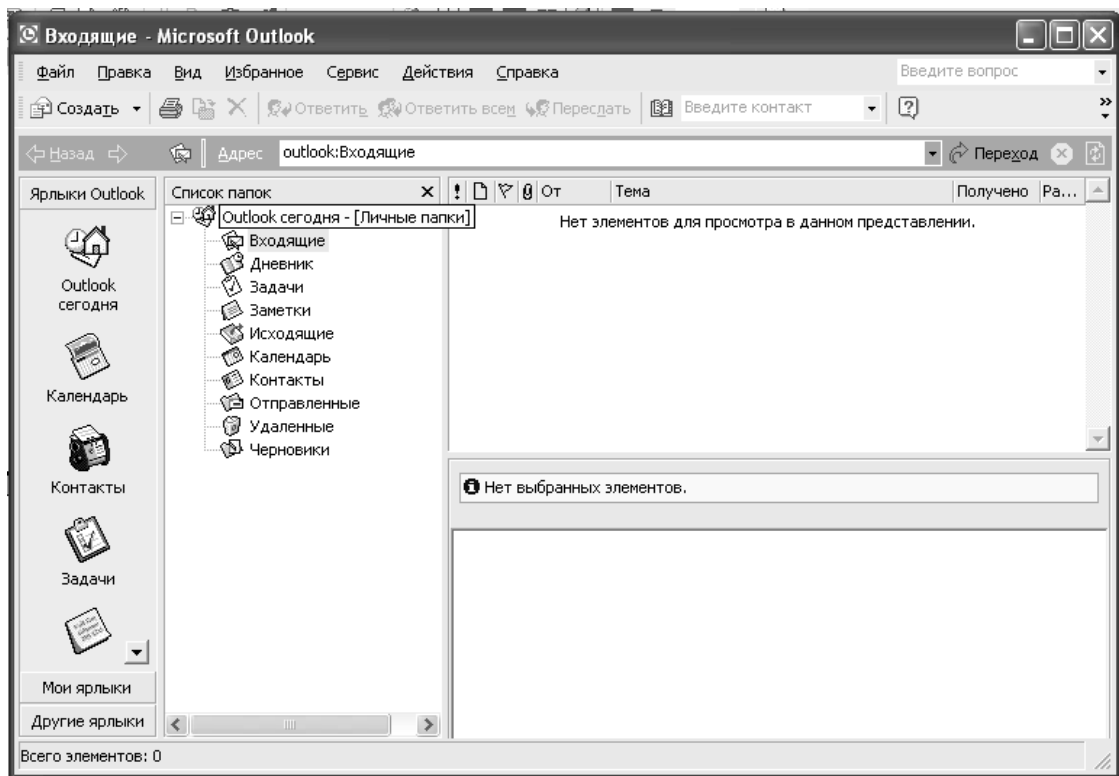


Рис.Окно почтовой программы MS Outlook

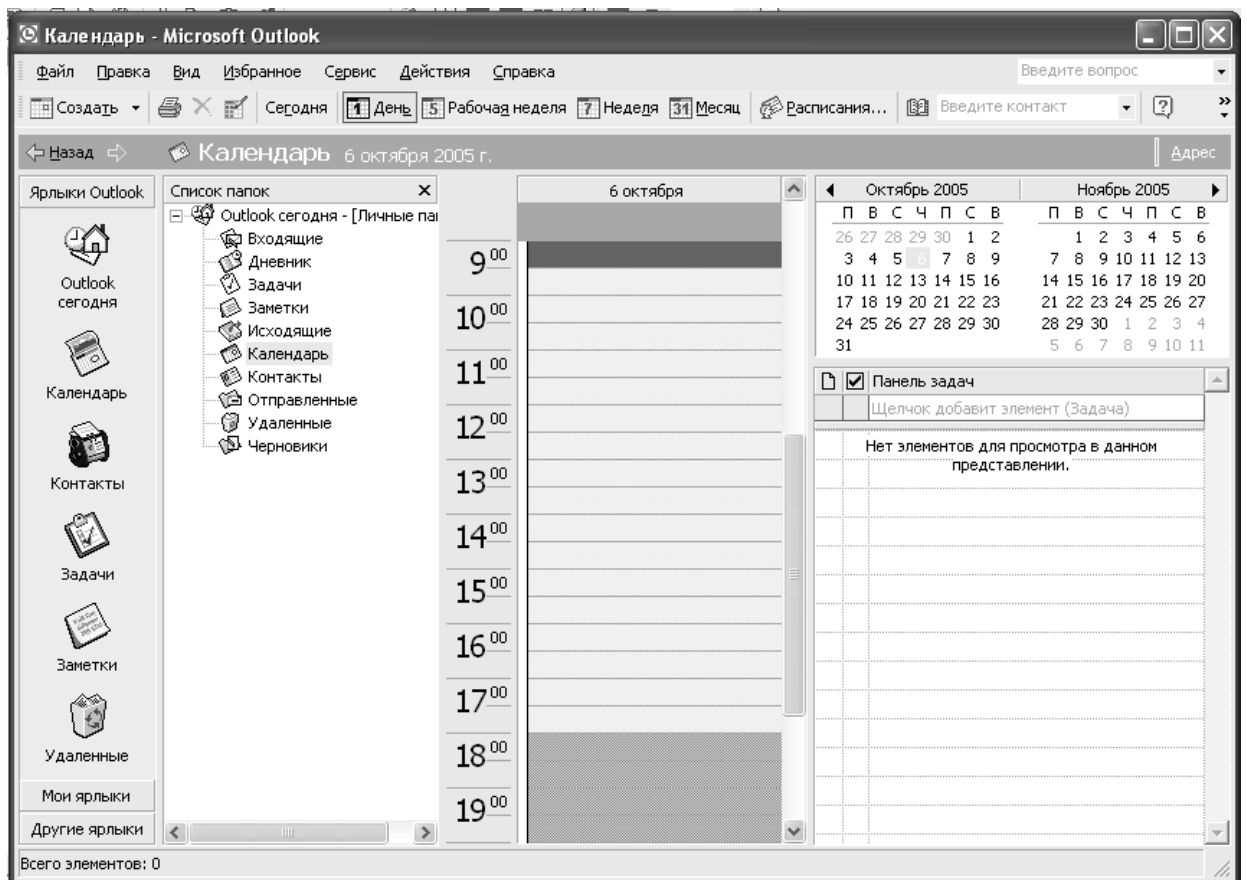


Рис. Календарь-планировщик задач в MS Outlook

2. Создать учетную запись в MS Outlook. Для этого ввести команду [Сервис – Учетные записи электронной почты]. Откроется диалоговая панель Учетные записи Интернета. Выбрать пункт Добавить новую учетную запись электронной почты. (Рис.40). Щелкнуть по кнопке Далее.

3. На появившейся диалоговой панели в поле **Тип сервера** для входящей почты: выберите POP3. Этот протокол наиболее часто используется для электронной почты.

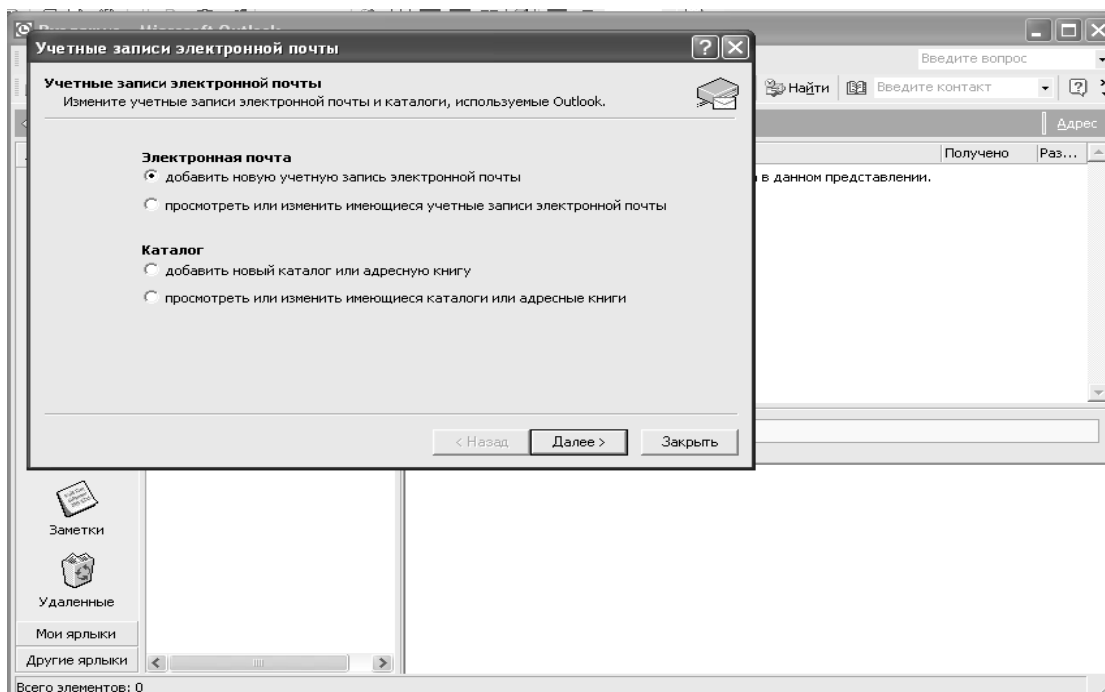


Рис. Добавление новой учетной записи электронной почты

4. Теперь необходимо указать имя почтового ящика и пароль для входа на почтовый сервер (Рис.41). В поле **Имя пользователя**: ввести имя (логин), которое вы указали при создании своего почтового адреса перед значком @. В поле **Пароль**: необходимо указать тот пароль, который был получен при регистрации почтового ящика.

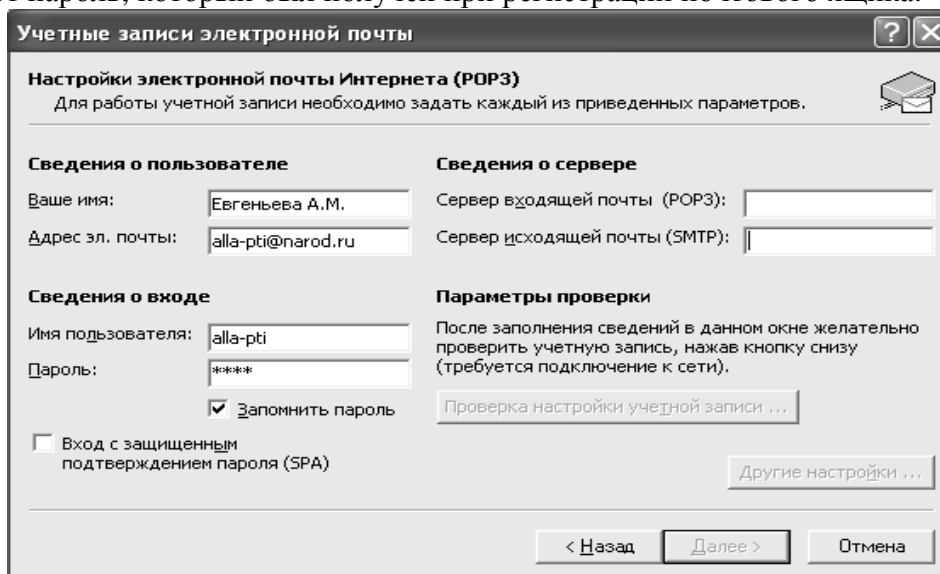
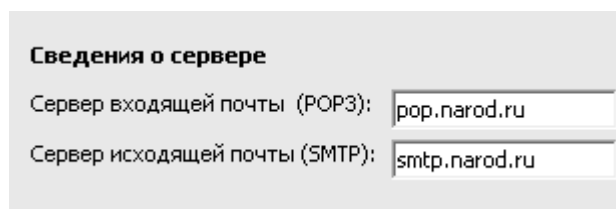


Рис. Создание учетной записи электронной почты



5. В поле **Ваше имя**: указать имя, которое будет видеть человек, получивший от вас письмо. **Адрес электронной почты**: указать тот адрес, который вы задали при регистрации подключения. Адрес следует указать целиком и именно в том виде, в котором вы его создали. Щелкнуть по кнопке Далее

6. В полях Сервер для входящей почты (POP3 или IMAP4): и Сервер для исходящей почты (SMTP): необходимо указать имена серверов входящей и исходящей почты, которые сообщает провайдер при регистрации подключения (Рис.). Щелкнуть по кнопке Далее



**Сведения о сервере**

Сервер входящей почты (POP3):

Сервер исходящей почты (SMTP):

Рис. Сведения о сервере на примере [www.narod.ru](http://www.narod.ru).

7. Нажать кнопку **Другие настройки** и на появившейся панели указать способ соединения с Интернетом, выбрать тип модема и используемое соединение с Интернетом.

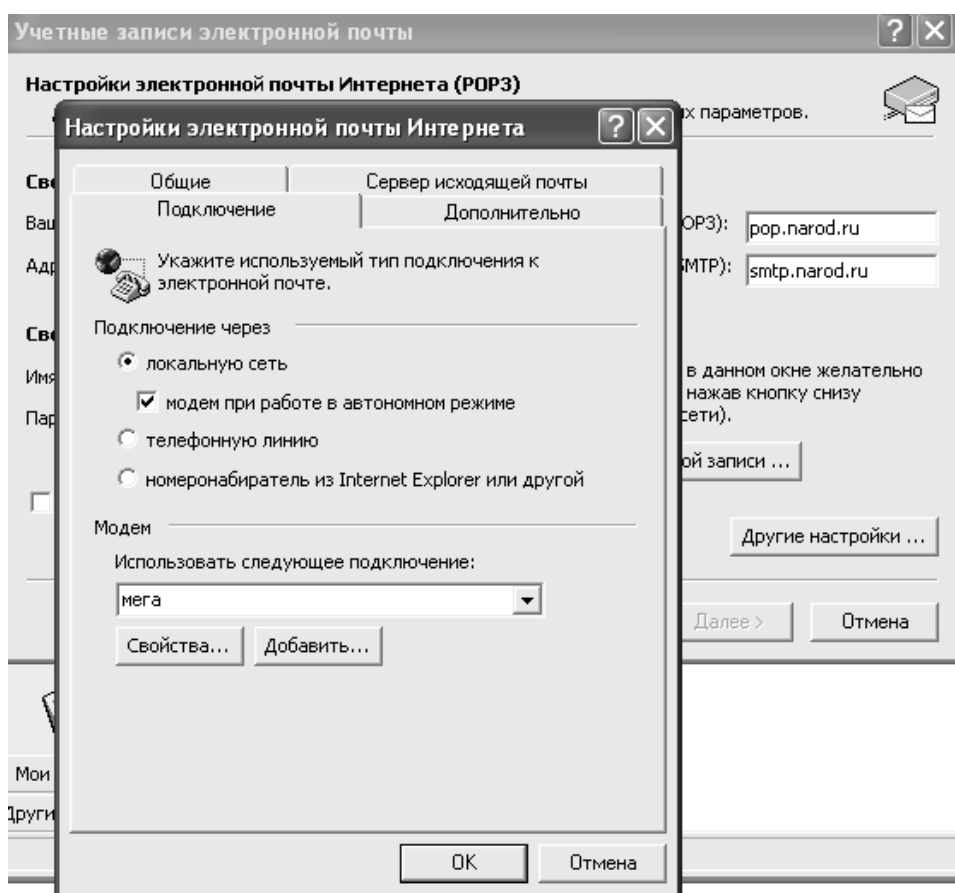


Рис. Настройки подключения к Интернет

8. Заданные выше параметры электронной почты объединяются вместе под одним именем — именем учетной записи. В поле Имя учетной записи почты сети Интернет: необходимо ввести имя для созданной учетной записи, например «ПочтаИнтернета». На рис.45 **Новая учетная запись** имеет имя АЛЛА.

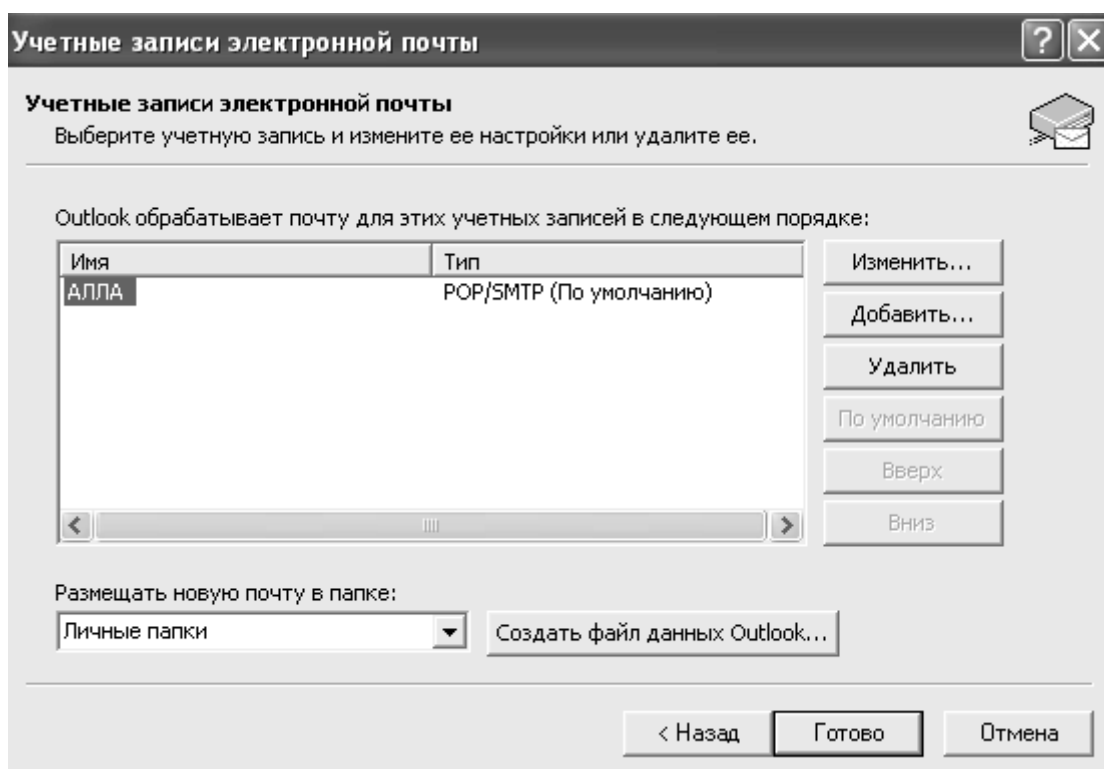


Рис. Готовая учетная запись

#### Задание №4. Создание, отправка и получение сообщения с помощью почтовой программы.

1. Ввести команду [Сообщение-Создать]. В окне *Создать сообщение* в поле *Кому:* необходимо указать электронный адрес адресата, например: *mirkro@mail.ru* В поле *Копии:* можно указать адреса получателей копии сообщения. В поле *Тема:* указывается тема сообщения, например «Проба».

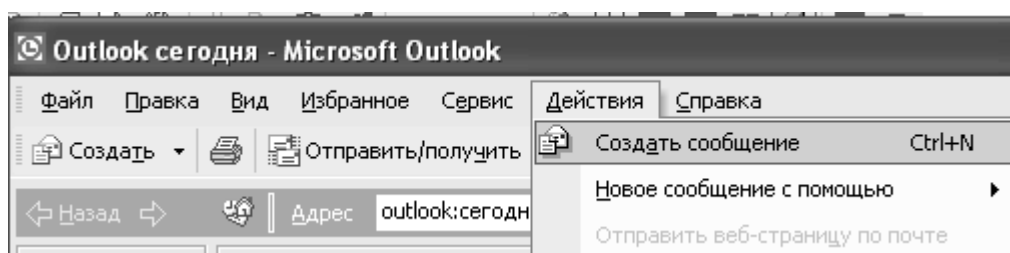


Рис. Создать новое сообщение

2. В области, отведенной для сообщения, вводится текст сообщения, например «Пробное сообщение (кодировкаКОИ8-Р)». Достаточно важен выбор правильной кодировки русских букв сообщения. При пользовании электронной почтой чаще всего используются кодировки Windows и КОИ8-Р

3. Выбор кодировки осуществить с помощью команды [Формат-Вид кодировки-Кириллица (КОИ8-Р)]. В сообщение можно вставлять файлы (текстовые, графические, звуковые и так далее).

4. Для вставки файла в сообщение необходимо ввести команду [Вставка-Вложение файла...]. В появившемся окне *Вставка* необходимо выбрать требуемый файл, и он будет вложен в сообщение (Рис).

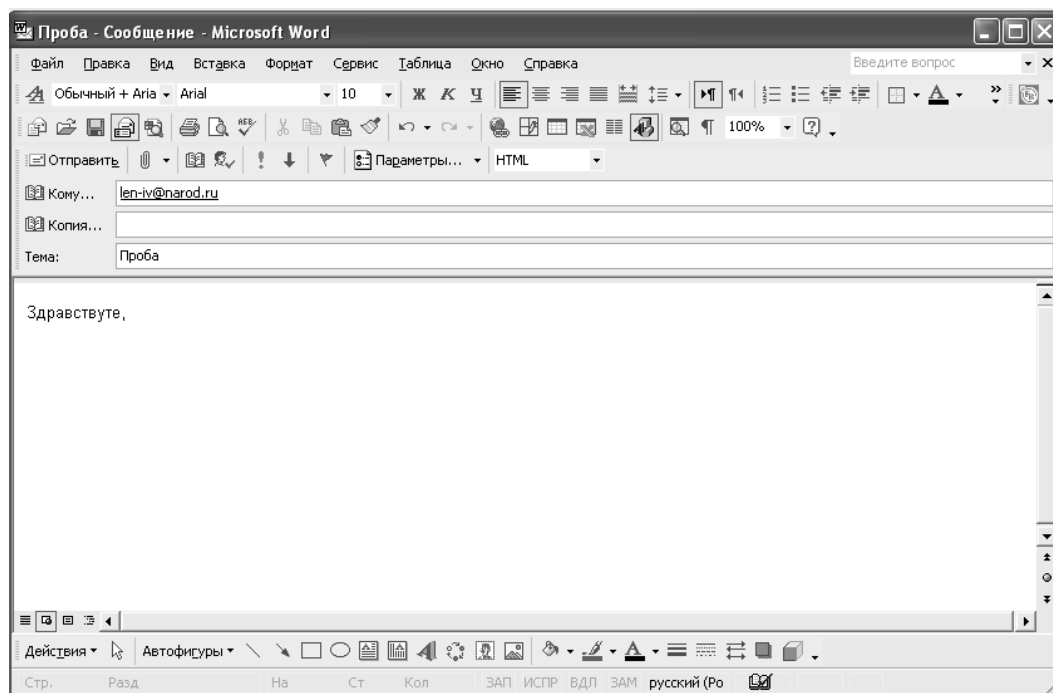


Рис. Интерфейс почтовой программы MS Outlook: окно для написания письма

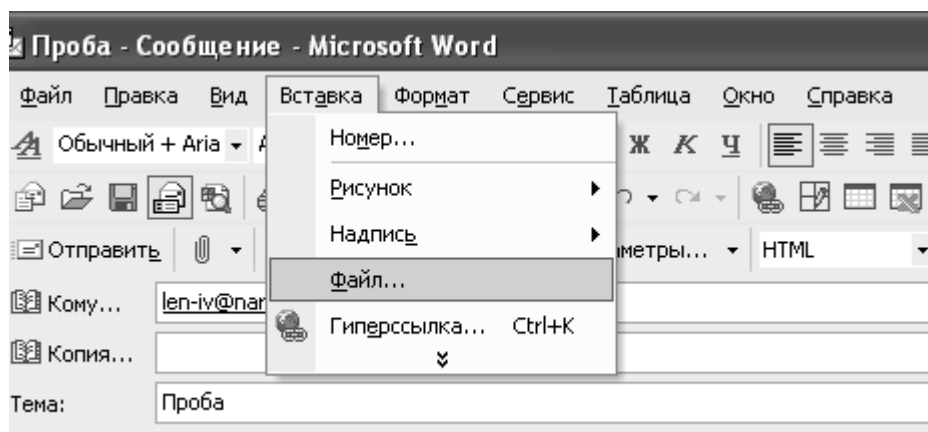


Рис. . Вложение файла в электронное письмо

1. Можно для вставки файла или документа воспользоваться на панели инструментов кнопкой с изображением скрепки. Любым способом вставим, например, в сообщение файл с презентацией из папки **Группа 287**, которая создана в лабораторной работе №3. Название вложенного файла появится в нижней части окна сообщения.

2. Если создание сообщения производилось в автономном режиме без подключения к Интернету, сообщение необходимо сохранить в папке Исходящие. После завершения работы над сообщением щелкнуть по кнопке **Отправить**, сообщение будет помещено в папку **Исходящие**

3. Для того чтобы отправить сообщение адресату, необходимо подключиться к Интернету. Щелкнуть по кнопке **Отправить почту**. Произойдет соединение с почтовым сервером, и все сообщения, находящиеся в папке **Исходящие** на локальном компьютере, будут доставлены на почтовый сервер. Одновременно отправленные сообщения будут перемещены на локальном компьютере в папку **Отправленные**. Почтовый сервер провайдера передаст сообщения в Интернет, и через некоторое время они будут доставлены на почто-

вые сервера получателей. В данном случае пробное сообщение попадет в почтовый ящик len-iv@narod.ru. Для получения сообщения абонент должен соединиться с Интернетом и произвести операцию доставки почты с почтового сервера провайдера на свой локальный компьютер.

4. Щелкнуть по кнопке *Отправить/получить почту*. В процессе доставки почты сообщения, хранящиеся в почтовом ящике на почтовом сервере, будут переданы на локальный компьютер получателя и размещены в папке *Входящие*. В случае установки кодировки, отличной от использованной при создании сообщения, сообщение будет представлять собой полную абракадабру. В этом случае необходимо подобрать "кодировку с помощью команды [Формат-Вид кодировки...].

### Самостоятельная работа

Научиться работать в телеконференциях.

В Приложении 12 подробно рассказано о назначении принципах работы служб Новости и Телеконференции. Для того чтобы иметь доступ к почтовому ящику какой-либо конференции, на нее необходимо «подписаться».

1. В окне почтовой программы Outlook Express выделить папку новостей (в данном случае *Конференции* и щелкнуть по кнопке *Группы новостей*).

2. Появится диалоговая панель *Группы новостей*, в которой высветятся названия десятков тысяч конференций. Для поиска интересующей тематики можно в поле *Показать группы новостей, содержащие:* задать символьный шаблон. Пусть, например, вас интересуют русскоязычные конференции, с большой долей вероятности их можно найти, если задать шаблон «rus». В окне откроется перечень конференций, в названии которых содержится этот шаблон. Следует выбрать некоторые из них (отметить с помощью мыши) и щелкнуть по кнопке *Подписаться*. Произойдет загрузка содержимого этих конференций (писем участников) на локальный компьютер пользователя.

3. Вы выбрали три интересующие вас конференции, которые теперь существуют в форме папок, вложенных в папку конференций *Конференции*. Выделите любую из них, например microsoft.public.ru.russian.outlookexpress (конференцию, посвященную проблемам работы в Outlook Express). В правом верхнем окне вы увидите ее содержание, то есть перечень присланных ранее сообщений.

4. Выделить интересующее вас сообщение, и в правом нижнем окне вы увидите содержание этого сообщения.

5. Ознакомившись с содержанием телеконференции, при желании можно послать в нее собственное сообщение, используя для создания бланка сообщения команду [*Сообщение-Ответить в группу новостей*].

### Лабораторная работа № 32 - Технология распознавания информации.

*Цель работы:* получить практические навыки по работе со сканером, программой оптического распознавания текста и оформлению текстовых документов, импортированных из других приложений в частности как распознанный текст со вставками. Научиться рационально выбирать параметры сканирования текстовой и графической информации.

*Должны знать:* основные команды программы по работе со сканером, основные команды программы Abby Fine Reader .

*Должны уметь:* правильно включать и выключать сканер, работать с мастером распознавания бумажных документов; включать принтер, проверять наличие картриджа и степень его заполнения чернилами; выбирать рациональные опции печати

#### Методические указания

➤ Программы для оптического распознавания текстов предназначены для распознавания бумажных документов всех видов и преобразования в электронный вид в форме информации заданного вида (текст, картинка, таблица). Как правило, профессиональные версии такого рода программ - это дорогостоящие программные продукты.

➤ В данной лабораторной работе предполагается использовать одно из наиболее удобных в работе и популярных приложений - ABBYY FINEREADER. Версия 7.0 HOME EDITION была выпущена в 2005 году. Более подробно о работе с этой программой, о шагах мастера сканирования и распознавания бумажных документов, о горячих клавишах изложено в ПРИЛОЖЕНИИ 7.

#### **Задание №1. Отсканировать документ, состоящий из двух страниц.**

Бумажный документ, подлежащий сканированию, состоит из двух страниц с текстом и вставками в виде рисунков, таблицы, формулы. Пример данного документа приведен в Приложении 8.

- Включить сканер, дать ему прогреться.
- Открыть крышку планшета и положить первый лист документа лицевой стороной вниз.
- Запустить программу Abby Fine Reader
- Выполнить первое действие Мастера: Сканировать
- В появившемся диалоговом окне мастера сканирования (программа-приложение к сканеру) выбрать параметры сканирования:

а) Плотность 300 или 600 точек на дюйм

б) Режим – оттенки серого (если рисунок должен остаться цветным, можно выбрать режим RGB). Но следует помнить, что цветная картинка займет места во много раз больше, чем черно-белая.

➤ Сделать предпросмотр файла, (команда Сканировать), на полученном в окне предпросмотра рисунке убедиться, что страница положена без излишнего наклона, на сканере формата А3 выбрать область сканирования, ограниченную лежащей на планшете страницей формата А4. Для этого область сканирования выделить мышкой.

➤ Отсканировать страницу окончательно, как файл с сохранением (команда Сканировать/Сохранить). В появившемся окне диалога подтвердить сохранение файла после сканирования в формате \*.jpg или \*.tiff, выбрав ему имя Лист1.

➤ Выполнить второе действие Мастера приложения Abby Fine Reader: *Разбить на блоки* (Сегментировать) документ. Всего на первой странице документа 5 сегментов (На рис.3 показана только часть листа для распознавания, но присутствуют все пять элементов). Следует отметить, что 7-я версия Abby Fine Reader автоматически разбивает документ на блоки, не фиксируя этот этап отдельно.

➤ Если программа не распознала элемент 4 как таблицу, то это надо сделать вручную. Правой кнопкой мыши вызвать контекстное меню и там выбрать команду **Распознать как / Таблица**.

➤ После сегментирования запустить третий шаг Мастера: *Распознать*. В результате с правой стороны окна программы заполняется окно с распознанным текстом.

➤ Сделать проверку полученного документа на наличие орфографических ошибок.

➤ Аналогичным образом распознать вторую страницу бумажного документа (сканировать, разбить на блоки, распознать).

➤ Импортировать полученный документ в приложение MS Word, где сохранить его под именем ПВМ.rtf. Шаблон \*. Rtf будет присвоен документу автоматически при импорте из приложения Abby Fine Reader.

#### **Самостоятельная работа**

■ Открыть сохраненный документ ПВМ.rtf и сохранить его в виде \*.doc. В качестве имени студенты могут выбрать любое название, например, свою фамилию.

■ Создать в документе *колонтитулы*: верхний с именем и фамилией, а так же названием лабораторной работы, нижний с номером страниц и количеством страниц в документе (2 страницы).

■ В меню **Файл** выбрать команду **Печать** и изучить Опции печати.

- Зайти в Меню Свойства принтера, выбрать Качество печати Нормальное. Проверить наличие чернил в черном и цветных картриджах. Для этого в Свойствах принтера, в меню сервис выбрать Панель инструментов принтера, где выбрать закладку *Предполагаемый уровень чернил*.

- Распечатать документ в виде одной копии документа так, чтобы на листе формата А4 поместилось 2 страницы. Для этого в диалоговом окне *Опции Печати* выбрать (справа) *Количество страниц на листе – 2*.

### 3.2. Методические рекомендации по выполнению курсовой работы

Курсовая работа предусмотрена для закрепления знаний по основам информатики и навыков программирования в среде Турбо Паскаль.

Пояснительная записка к курсовой работе выполняется (набирается) на компьютере в текстовом редакторе WORD.

Курсовая работа содержит необходимые в соответствии с требованиями нормоконтроля разделы.

#### Структура курсовой работы

Введение (указываются цели и задачи курсовой работы)

1 Стандартный модуль Crt.

2 Стандартный модуль Graph.

3 Программирование на языке Turbo Pascal

...

3.1 Создание программ для расчета значения функции с заданным диапазоном и шагом изменения аргумента

3.2 Программа с использованием оператора цикла с постусловием

3.3 Программа с использованием оператора цикла с предусловием

3.4 Программа с использованием процедур и функций модуля Crt

4 Создание программ для расчета статистических характеристик

4.1 Программа для расчета статистических характеристик с файловым вводом данных

4.2 Программа для расчета статистических характеристик с файловым выводом данных

4.3 Программа для расчета статистических характеристик с использованием процедур и функций модуля Graph

4.4 Создание программы с использованием условных операторов

4.5 Создание программы с использованием оператора цикла FOR

5 Создание программы для реализации процесса тестирования ...

5.1 Программа тестирования с вопросами, представленными в файле исходных данных

5.2 Программа тестирования с вопросами, представленными в программе

5.3 Программа тестирования с использованием процедур и функций модуля Crt

Заключение (делаются выводы по работе)

Список использованных источников

Приложения (приводятся распечатки всех программ)

#### ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

№ недели	Дата	Раздел курсовой работы
1,2	1.09-10.09	Выдача задания к курсовой работе
3,4	11.09-23.09	1. Стандартный модуль Crt
5,6	25.09-7.10	2. Стандартный модуль Graph
		Контрольная точка 1
7,8	9.10-21.10	3.1. Создание программ для расчета значения функции с заданным диапазоном и шагом изменения аргумента
9,10	23.10-4.11	3.2 Создание программ для расчета статистических характеристик
		Контрольная точка 2

11,12	6.11-18.11	3.3 Создание программы с использованием условных операторов 3.4. Создание программы с использованием оператора цикла FOR
13,14,15	20.11-9.12	5. Создание программы для реализации процесса тестирования
		Контрольная точка 3
16	11.12-16.12	Оформление курсовой работы
17,18	18.12-30.12	Защита курсовой работы

### 3.3. Методические указания по самостоятельной работе студентов

Объем самостоятельной работы студентов определяется учебным планом.

Для освоения курса «Информатика предусмотрено, что 50 % времени от аудиторных занятий студент должен заниматься самостоятельно. В этой связи студент должен планировать свое время, учитывая, что он наряду с данной дисциплиной должен изучать и другие, входящих в учебный план подготовки бакалавров по направлению «Сервис». Таким образом, для получения полноценного образования по сервису студент должен проявлять желание и много трудиться, чтобы стать высококвалифицированным специалистом.

Цель самостоятельной работы студента заключается в глубоком, полном освоении учебного материала и в развитии навыков самообразования. В целом разумное сочетание самостоятельной работы с иными видами учебной деятельности позволяет реализовать три основных компонента университетского образования:

- познавательный, который заключается в усвоении студентами необходимой суммы знаний по избранной специальности, а также способности самостоятельно их пополнять;
- развивающий, то есть выработка навыков аналитического и логического мышления, способности профессионально оценить ситуацию и найти правильное решение;
- воспитательный – формирование профессионального сознания, мировоззренческих установок, связанных не только с выбранной им специальностью, но и с общим уровнем развития личности.

Самостоятельная работа по выполнению домашних заданий представляет собой логическое продолжение аудиторных занятий, проводится по заданию преподавателя, который инструктирует студентов и устанавливает сроки выполнения задания. Режим и продолжительность работы выбирает сам студент в зависимости от своих способностей и конкретных условий, что требует от него не только умственной, но и организационной самостоятельности.

Самостоятельная работа студентов складывается из нескольких составляющих:

- работа с текстами: учебниками и дополнительной литературой в виде учебных пособий, методических рекомендаций, научных книг и статей;
- просмотр конспектов лекций; написание отчетов по лабораторным работам;
- подготовка к зачету, экзамену.

Самостоятельная работа студентов по дисциплине включает:

- самостоятельное изучение теоретических разделов дисциплины по заданию лектора;
- повторение и углубленное изучение лекционного материала;
- написание реферата;
- подготовку и выполнение лабораторных работ;
- выполнение индивидуальных заданий, контрольных работ и расчетов на ПК;
- подготовку к зачету и экзамену.

*Темы для самостоятельного изучения:*

- 1) Кодирование информации.

- 2) Двоичная система счисления.
- 3) История развития вычислительных средств.
- 4) Аппаратура компьютера.
- 5) Архитектура компьютера.
- 6) Системное и прикладное программное обеспечение.
- 7) Понятие об операционной системе.
- 8) Прикладное программное обеспечение.
- 9) Машинная графика. Назначение и основные функции.
- 10) Системы управления базами данных (СУБД). Назначение и основные функции.
- 11) Электронные таблицы. Назначение и основные функции.
- 12) Понятие программного продукта. Жизненный цикл программного обеспечения.
- 13) Задачи, решаемые с помощью баз данных.
- 14) Экономические, организационные и правовые вопросы создания программного и информационного обеспечения. Понятие интеллектуальной собственности.
- 15) Понятие телекоммуникации. Компьютерные сети как средство реализации практических потребностей.
- 16) Интегрированные автоматизированные системы
- 17) Организационные меры, инженерно-технические и иные методы защиты информации в том числе сведений, составляющих государственную тайну.
- 18) Угрозы безопасности информации и их классификация. Основные виды защищаемой информации. Проблемы ИБ в мировом сообществе.
- 19) Интеллектуальные системы решения вычислительных задач и моделирования.
- 20) Искусственный интеллект и интеллектуальные системы.

Тематика работ для самостоятельного изучения сообщается студентам на первой лекции. Материал может быть подготовлен в виде реферата и (или) представлен в виде доклада на соответствующей лекции или лабораторном занятии.

## **4 КОНТРОЛЬ ЗНАНИЙ**

### **4.1. Перечень форм контроля**

Промежуточный контроль знаний студентов осуществляется при подготовке к работе, выполнении и сдаче каждого задания лабораторной работы, а так же во время контрольных точек при выполнении курсовой работы.

В качестве заключительного контроля знаний студентов в 1 семестре служит зачет. На зачете оцениваются знания студентов по выполненным лабораторным работам. Во 2 семестре заключительным контролем служит экзамен, в 3 семестре – курсовая работа. К экзамену допускаются студенты при выполнении и защите всех лабораторных работ.

### **4.2. Оценка знаний студентов**

Нормы оценки знаний предполагают учет индивидуальных особенностей студентов, дифференцированный подход к обучению, проверке знаний, умений.

В устных ответах студентов на экзамене и при защите курсовой работы учитываются: глубина знаний, полнота знаний и владение необходимыми умениями (в объеме полной программы); осознанность и самостоятельность применения знаний и способов учебной деятельности, логичность изложения материала, включая обобщения, выводы (в соответствии с заданным вопросом), соблюдение норм литературной речи.

### **4.3. Критерии оценки**

Оценка знаний на экзамене и при защите курсовой работы производится по четырех балльной системе.

Оценка "пять" – материал усвоен в полном объеме; изложен логично; основные умения сформулированы и устойчивы; выводы и обобщения точны.

Оценка "четыре" – в усвоении материала незначительные пробелы, изложение недостаточно систематизированное; отдельные умения недостаточно устойчивы; в выводах и обобщениях допускаются некоторые неточности.

Оценка "три" – в усвоении материала имеются пробелы: материал излагается не-



систематизированно; отдельные умения недостаточно сформулированы; выводы и обобщения аргументированы слабо; в них допускаются ошибки.

Оценка "два" – основное содержание материала не усвоено, выводов и обобщений нет.

#### **4.4 Подготовка к экзамену**

Экзамен – форма итоговой проверки знаний студентов. Особенность экзамена состоит в том, что для него дается время на подготовку (в отличие от зачета). Поэтому для успешной сдачи экзамена нужно активно работать на лабораторных занятиях, выполнить все контрольные задания и до зачетной недели выучить теоретический материал. Для успешного запоминания большого объема информации рекомендуется соблюдать несколько правил:

- подготовка к экзамену должна проводиться систематически, в течение всего семестра;

- интенсивная подготовка должна начаться накануне зачетной недели, т.е. распределить контрольные вопросы таким образом, чтобы успеть выучить их полностью до экзамена;

- время, которое выделяется для сдачи экзамена, надо использовать для повторения материала.

##### *Перечень экзаменационных вопросов*

- 1) Понятие информации. Свойства информации. Место и роль понятия «информация» в курсе информатики.
- 2) Кодированная информация.
- 3) Понятие носителя информации. Формы представления и передачи информации.
- 4) Назначение компьютера.
- 5) История развития вычислительных средств.
- 6) Основные функциональные части компьютера.
- 7) Основные принципы работы компьютера.
- 8) Программа как последовательность действий компьютера. Понятие о машинном языке и языке Ассемблер.
- 9) Аппаратура компьютера.
- 10) Архитектура компьютера.
- 11) Центральные устройства компьютера.
- 12) Понятие алгоритма и алгоритмической системы.
- 13) Линейные, разветвленные и циклические алгоритмы. Вложенные и параллельные алгоритмы.
- 14) Компьютер как исполнитель алгоритмов.
- 15) Программа как изображение алгоритма в терминах команд, управляющих работой компьютера.
- 16) Коды, ассемблеры, языки высокого уровня. Трансляция и компоновка.
- 17) Программное обеспечение компьютера.
- 18) Системное и прикладное программное обеспечение.
- 19) Понятие об операционной системе.
- 20) Прикладное программное обеспечение.
- 21) Понятие языка высокого уровня. Синтаксис и семантика. Элементы и структуры данных, алфавит, имена, выражения, операции, операторы языка программирования Паскаль
- 22) Структура программ, аппарат подпрограмм, реализация логических структур языка программирования Паскаль.
- 23) Понятие программного продукта. Жизненный цикл программного обеспечения.
- 24) Задачи, решаемые с помощью баз данных.
- 25) Понятие телекоммуникации. Компьютерные сети как средство реализации практических потребностей.

- 26) Локальные сети и глобальные сети: принципы построения, архитектура, основные компоненты, их назначение и функции.
- 27) Всемирная компьютерная сеть ИНТЕРНЕТ. Ее возможности.
- 28) Модели решения функциональных и вычислительных задач.
- 29) Компьютерная графика
- 30) Системы геометрического моделирования.
- 31) Интегрированные автоматизированные системы
- 32) Угрозы безопасности информации и их классификация. Основные виды защищаемой информации. Проблемы ИБ в мировом сообществе.
- 33) Организационные меры, инженерно-технические и иные методы защиты информации, в том числе сведений, составляющих государственную тайну.
- 34) Защита информации в локальных компьютерных сетях, антивирусная защита.

#### **4.5 Рейтинговая система контроля знаний**

##### *Порядок определения рейтинговой оценки по учебной дисциплине*

Каждая учебная дисциплина независимо от ее общей трудоемкости, оценивается в баллах. Учебная деятельность студента оценивается по 100-балльной шкале.

Все преподаваемые в университете дисциплины по итоговой форме контроля знаний делятся на две категории:

- с экзаменом;
- без экзамена (с зачетом).

В университете действует следующая шкала пересчета рейтинговых оценок. По дисциплине с экзаменом или дифференцированным зачетом границы оценки задаются следующим образом:

- менее 51 балла – «неудовлетворительно»;
- от 51 до 74 баллов – «удовлетворительно»;
- от 75 до 90 баллов – «хорошо»;
- от 91 до 100 баллов – «отлично».

Для дисциплин без экзамена (с зачетом) оценка «зачтено» ставится по нижней границе оценки «удовлетворительно», т.е. начиная с 51 балла.

Студент, получивший положительные оценки по всем дисциплинам (курсовым работам) основной образовательной программы в течение семестра, считается успешно выполнившим основную образовательную программу и продолжает дальнейшее обучение в соответствии с графиком учебного процесса.

В случае, если качество учебных работ осталось неудовлетворительным, или рейтинговая оценка студента за семестр ниже 15 баллов, студент представляется к отчислению.

Студент, пропустивший занятия при наличии уважительной причины, имеет право повысить свой рейтинговый балл.

*Методика формирования балльно-рейтинговой оценки по дисциплине «ИНФОРМАТИКА» (осенний семестр: форма отчетности - зачет)*

Максимальная сумма баллов за текущий контроль - **80** баллов.

Максимальная сумма баллов, набранных на зачете, - **20** баллов.

Итого **100** баллов.

Рейтинговая оценка знаний является интегральным показателем качества теоретических знаний и навыков студентов по курсу и складывается из следующих компонентов:

- а) работа на лекциях (выполнение самостоятельных и домашних заданий);
- б) выполнение тестовых и контрольных заданий для текущего контроля;
- в) промежуточная аттестация;
- г) выполнение контрольной работы;
- д) зачет

Суммарный рейтинг, необходимый для получения зачета, определяется следующим образом:

- работа на лекциях: 10 баллов (по 1 баллу за выполненную работу на каждой лекции плюс 2 балла в конце курса для студентов, не пропустивших ни одной лекции без уважительной причины. Для студентов, пропускающих лекции без уважительной причины, за каждую пропущенную лекцию из общей суммы рейтинга вычитается 2 балла);

- работа на лаб.раб.: 50 баллов (за выполнение и сдачу лаб.раб. в соответствии с календарным планом лаб.раб.);

- выполнение контрольных работ (2 контрольные работы): 8 баллов (по 4 балла за каждую контрольную работу по 4-х балльной системе);

- промежуточная аттестация (2 контрольной точки): 10 баллов (по 5 баллов за каждую в соответствии с текущим рейтингом).

*Бонусы* (поощрительные баллы) студент получает к своему рейтингу в конце семестра:

- за посещение всех лекционных занятий: 2 балла;

- за активную и регулярную работу на лекциях и лаб.раб.: 2 балла;

- при выполнении и защите раньше срока лабораторных работ: по 2 балла за каждую работу.

*Штрафы* (штрафные баллы, которые вычитаются из текущего рейтинга):

- за пропуск без уважительной причины лекции: **2 балла** за 1 лекцию;

- за пропуск без уважительной причины лабораторной работы: **2 балла** за 1 лабораторную работу;

- за сдачу позднее установленного срока лабораторных работ: **2 балла** за первую неделю просрочки, **1 балл** за каждую следующую неделю.

Минимальное значение рейтинговой оценки, набранной студентом по результатам текущего контроля по всем видам занятий, при которой студент допускается к сдаче зачета, составляет **51** баллов.

Если к моменту проведения зачета студент набирает **71** и более баллов, оценка может быть выставлена ему в ведомость и в зачетную книжку без процедуры принятия зачета. Выставление оценок производится на последней неделе теоретического обучения по данной дисциплине.

Устранение задолженности по отдельным контролируемым темам дисциплины в рамках текущего контроля может проводиться в форме дополнительного контрольного опроса по материалу тем дисциплины, по которым студент желает повысить балл. Дополнительный контрольный опрос по этим темам проводится в течение теоретического обучения преподавателем, который проводил занятия с данными студентами. По желанию студента ему дается право не более одного раза пройти дополнительный контрольный опрос по этим темам.

Устранение задолженности студента по отдельным контролируемым темам дисциплины может проходить в течение семестра в часы дополнительных занятий или консультаций, установленных в расписании по соответствующим видам занятий по данной дисциплине.

Устранение задолженности по текущему контролю для допуска студента к зачету проводится на последней неделе теоретического обучения по данной дисциплине.

Для студентов, пропустивших более  $\frac{1}{2}$  занятий по дисциплине, сдача зачета является обязательной. Для студентов, пропустивших более 30 % занятий по болезни (подтверждается медицинской справкой), и для студентов, пропустивших занятия без уважительной причины, либо получивших на занятии неудовлетворительную оценку, отработка пропущенного занятия является обязательной. При этом полученная оценка в текущий рейтинг не включается.

*Методика формирования балльно-рейтинговой оценки по дисциплине* (весенний семестр - экзамен)

Максимальная сумма баллов за текущий контроль (*текущий рейтинг*) - **60** баллов.

Максимальная сумма баллов, набранных на экзамене (*теоретический рейтинг*) - **40** баллов.

Итого

- **100** баллов.

Минимальное значение рейтинговой оценки, набранной студентом по результатам текущего контроля по всем видам занятий в течение семестра, при которой студент допускается к сдаче экзамена, составляет **40** баллов.

Рейтинговая система предусматривает поощрение студентов за хорошую работу в семестре проставлением «премиальных» баллов(бонусов) с возможностью получить семестровую оценку без сдачи экзаменов или зачетов.

Выставление оценок за экзамен производится в период сессии.

Устранение задолженности по отдельным контролируемым темам дисциплины в рамках текущего контроля может проводиться в форме дополнительного контрольного опроса по материалу тем дисциплины, по которым студент имеет право повысить балл.

Устранение задолженности студента по отдельным контролируемым темам дисциплины может проходить в течение семестра в часы дополнительных занятий или консультаций, установленных в расписании по соответствующим видам занятий данной дисциплины.

Студент, набравший к моменту окончания семестра **менее 40** баллов по текущему контролю, считается не выполнившим график учебного процесса, аттестуется по дисциплине неудовлетворительно и к экзамену не допускается.

Студент, получивший по результатам текущего контроля и экзамена рейтинговую оценку по дисциплине **менее 51 балла**, аттестуется неудовлетворительно и ему предоставляется возможность ликвидировать задолженность по дисциплине в установленном порядке (согласно положению о курсовых экзаменах и зачетах).

Состав рейтинговой системы оценки:

**Текущий рейтинг** (за работу в течение семестра) – **60 баллов**, из них;

- *работа на лекциях*: **10 баллов** (по 1 баллу за выполненную работу на каждой лекции плюс 2 балла в конце курса для студентов, не пропустивших ни одной лекции без уважительной причины. Для студентов, пропускающих лекции без уважительной причины, за каждую пропущенную лекцию из общей суммы рейтинга вычитается 2 балла);

- *выполнение лабораторных работ*: **30 баллов** (за выполнение и сдачу лабораторных работ в соответствии с календарным планом лабораторных работ);

- *выполнение контрольных работ* (2 контрольные работы): **10 баллов** (по 5 баллов за каждую контрольную работу по 5-ти балльной системе);

- *промежуточная аттестация* (2 контрольной точки): **10 баллов** (по 5 баллов за каждую в соответствии с текущим рейтингом).

**Бонусы** (поощрительные баллы) студент получает к своему рейтингу в конце семестра:

- за посещение всех лекционных занятий: **2 балла**;

- за активную и регулярную работу на лекциях и лаб. раб.: **2 балла**;

- при выполнении и защите раньше срока лабораторных работ: по **2 балла** за каждую работу.

**Штрафы** (штрафные баллы, которые вычитаются из текущего рейтинга):

- за пропуск без уважительной причины лекции: **2 балла** за 1 лекцию;

- за пропуск без уважительной причины лабораторной работы: **2 балла** за 1 лабораторную работу;

- за сдачу позднее установленного срока лабораторных работ: **2 балла** за первую неделю просрочки, **1 балл** за каждую следующую неделю.

**Теоретический рейтинг** (баллы полученные студентом при сдаче экзамена) составляет **40 баллов**, из них:

- 1 экзаменационный вопрос: **10 баллов**;

- 2 экзаменационный вопрос: **10 баллов**;

- задача:

**20 баллов.**

**Структура рейтинг-плана**  
(осенний семестр)

Таблица 1 - Соотношение видов рейтинга

№ п\п	Вид рейтинга	Весовой коэффициент, %	Максимальное количество баллов
1	текущий	80	80
2	теоретический	20	20
	Итого	100	100

Таблица 2 - Соотношение видов учебной деятельности студентов в рамках текущего рейтинга

№ п\п	Вид учебной деятельности	Весовой коэффициент, %	Максимальное количество баллов
1	Посещение занятий	2,5	2
2	Работа на лекциях	12,5	10
3	Выполнение и защита лаб. работ	62,5	50
4	Контрольная работа	10	8
5	Промежуточная аттестация	12,5	10
	Итого	100	80

Таблица 3- Календарный план мероприятий по дисциплине

№ п\п	№ учебной недели	Название блока (темы, модуля)	Вид контроля	Максимальное количество баллов
1	1	Аппаратное обеспечение компьютера	Конспект Защита лаб.раб. 1	2
2	2,3	Текстовый редактор и текстовый процессор	Контрольная работа опрос	4
3	4 - 5	Работа с файлами и папками: создание, удаление, поиск и сортировка	Индивидуальные задания Тест	4
4	6 - 8	Работа с текстовым редактором при форматировании текста	Опрос Индивид. задание	5
5	9 - 10	Работа с текстовым редактором при создании таблиц	Опрос Защита лаб.раб. индивидуальное задание	5
6	11,12	Работа с текстовым редактором при создании и редактировании формул	Защита лаб.раб, индивидуальное задание опрос	7
7	13	Работа с текстовым редактором при создании графических объектов	индивидуальное задание	3
8	14-15	Автоформализация профессиональных знаний на экране компьютера с помощью графического редактора	Контрольная работа Индивидуальное задание	10
9	16 - 18	Проектирование визуального		10

		алгоритма на экране компьютера		
Итого:				50

### Структура рейтинг-плана (весенний семестр)

Таблица 5 - Соотношение видов рейтинга

№ п\п	Вид рейтинга	Весовой коэффициент, %	Максимальное количество баллов
1	текущий	60	60
2	теоретический	40	40
	Итого	100	100

Таблица 6 - Соотношение видов учебной деятельности студентов в рамках текущего рейтинга

№ п\п	Вид учебной деятельности	Весовой коэффициент, %	Максимальное количество баллов
1	Посещение занятий	3	2
2	Работа на лекциях	13	8
3	Выполнение и защита лаб. работ	50	30
4	Контрольная работа	17	10
5	Промежуточная аттестация	17	10
	Итого	100	60

Таблица 7 - Календарный план мероприятий по дисциплине

№ п\п	№ учебной недели	Название блока (темы, модуля)	Вид контроля	Максимальное количество баллов
1	1,2,3	Языки программирования. Язык программирования Паскаль. Стандартные типы данных в Паскале.	Защита лаб.раб. 1, конспект 1	4
2	4,5	Структура программы в Паскаль	Защита лаб.раб. 2, конспект 2	4
3	6,7,8	Ввод-вывод данных в Паскале	Защита лаб.раб. 3, индивидуальное задание 1	5
4	9,10	Файловый ввод данных в Паскале	Защита лаб.раб. 4, индивидуальное задание 2	4
5	11,12	Операторы языка Паскаль	Опрос Защита лаб.раб. 5, индивидуальное задание 3 Конспект 3	4
6	13,14	Файловый вывод данных в Паскале	Защита лаб.раб. 6, индивидуальное задание 4, опрос	4
7	15,16,17	Работа с символами и строковыми данными	Защита лаб.раб. 7, индивидуальное задание 5 Конспект 4	5
Итого:				30

## 5. Интерактивные технологии и инновационные методы, используемые в образовательном процессе

Для реализации компетентного подхода программа дисциплины «Информатика» предусматривает использование в учебном процессе активных и интерактивных форм проведения занятий.

Одним из элементов педагогической системы являются организационные формы обучения. Они делятся на индивидуальные, коллективные групповые, аудиторные и вне-аудиторные. Эта классификация не является строго научной, но позволяет несколько упорядочить разнообразие форм обучения.

Организационные формы обучения – это виды учебных занятий, отличающихся друг от друга дидактическими целями, местом проведения, продолжительностью.

В рамках различных организационных форм обучения преподаватель обеспечивает активную познавательную деятельность обучаемых, используя фронтальную, групповую и индивидуальную работу.

Фронтальная работа предполагает совместную деятельность всей группы: преподаватель излагает материал для всей группы, ставит одинаковые задачи, а студенты решают одну проблему.

При групповой работе учебная группа разделяется на несколько подгрупп, которые выполняют одинаковые или различные задания.

При индивидуальной работе каждый студент получает задание, которое он выполняет независимо друг от друга. Индивидуальная форма организации учебного труда особенно целесообразна для таких видов работ, в которых могут более ярко проявиться индивидуальные особенности и возможности обучаемых.

*Пример индивидуального задания.*

Тема задания - **Работа с электронной почтой и в телеконференциях**

**Цель работы:** получить практические навыки по работе с электронной почтой и в телеконференциях.

За последнее время для работы с электронной почтой стали часто использоваться web-технологии. Появились web-сайты, которые предлагают всем желающим зарегистрировать свой бесплатный электронный почтовый ящик. Примером таких сайтов могут являться <http://mail.ru>, <http://www.rambler.ru>, <http://www.yandex.ru> Для работы с такой электронной почтой не требуются специальные почтовые программы, т.к можно воспользоваться любым браузером для загрузки отображения соответствующей web-страницы.

Группа делится на несколько подгрупп (3 – 4 человека) каждая подгруппа получает индивидуальное задание. В течение часа группа решает задачу и затем предлагает презентацию своего решения

Для выполнения задания студент:

- **должны знать:** правила пользования электронной почтой, этику общения в телеконференциях по темам; как создать почтовый электронный ящик на бесплатном российском сервере, как установить почтовую программу на примере MS Outlook;

- **должны уметь:** правильно работать с программой Outlook, выполнять основные установки, в том числе коммутацию с Интернет, создавать электронный почтовый ящик, настраивать его.

## СОДЕРЖАНИЕ

1	Рабочая программа	3
1.1	Цели и задачи освоения дисциплины	3
1.2	Место дисциплины в структуре ОПП	3
1.3	Требования к результатам освоения дисциплины	3
1.4	Объем дисциплины и виды учебной работы	4
1.5	Содержание разделов и тем дисциплины	4
1.6	Самостоятельная работа	6
1.7	Курсовая работа	7
1.8	Образовательные технологии	8
1.9	Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и Учебно-методическое обеспечение самостоятельной работы студентов	9
1.10	Учебно-методическое и информационное обеспечение дисциплины	12
1.11	Материально-техническое обеспечение дисциплины	13
2	Краткое изложение программного материала	13
3	Методические указания по изучению дисциплины	30
3.1	Методические указания для преподавателя	30
3.2	Методические указания к лабораторным работам	32
3.3	Методические указания по выполнению курсовой работы	70
3.4	Методические указания по самостоятельной работе	71
4	Контроль знаний	72
4.1	Перечень форм контроля	72
4.2	Оценки знаний студентов	72
4.3	Критерии оценки	72
4.4	Итоговый контроль знаний	73
4.5	Рейтинговая система контроля знаний	74
5	Интерактивные технологии и инновационные методы, используемые в образовательном процессе	79