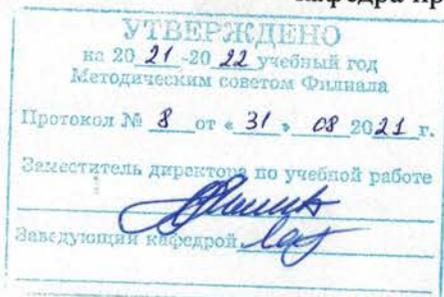


Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  
Московский государственный университет имени М.В. Ломоносова  
филиал МГУ в г. Севастополе  
факультет компьютерной математики  
кафедра программирования



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Наименование дисциплины (модуля):**

*ПРАКТИКУМ НА ЭВМ*

*код и наименование дисциплины (модуля)*

**Уровень высшего образования:**  
*бакалавриат*

**Направление подготовки:**

*01.03.02 Прикладная математика и информатика*

*(код и название направления/специальности)*

**Направленность (профиль) ОПОП:**  
*общий*

*(если дисциплина (модуль) относится к вариативной части программы)*

**Форма обучения**

*очная*

Рабочая программа рассмотрена  
на заседании кафедры программирования  
протокол № 3 от « 28 » апреля 2020 г.  
Руководитель ОП 01.03.02 «Прикладная  
математика и информатика»  
*(подпись)* (Н. В. Лактионова)

Рабочая программа одобрена  
Методическим советом  
Филиала МГУ в г.Севастополе  
Протокол № 6 от « 10 » июня 2020 г.  
*(подпись)* (А.В. Мартынкин)

Севастополь, 2020

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки 01.03.02 «Прикладная математика и информатика» в редакции приказа МГУ от 30 декабря 2016 г.

*курс – I, III, IV*

*семестры – 1, 2, 5, 6, 7*

*зачетных единиц – 12*

*академических часов – 432, в т.ч.:*

*лекций – нет*

*практических занятий – 252*

*Формы промежуточной аттестации:*

*зачёты с оценкой в семестрах 1, 2, 5, 6*

*экзаменов - нет*

*Форма итоговой аттестации:*

*зачёт с оценкой в семестрах 7*

**ОГЛАВЛЕНИЕ**

ВВЕДЕНИЕ .....	3
1. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	8
2. СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ .....	13
3. РЕКОМЕНДУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ .....	17
5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ .....	29
6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	32
ПРИЛОЖЕНИЕ А.....	33

## ВВЕДЕНИЕ

### **1. Цель и задачи освоения дисциплины**

Целью данного курса является практическое освоение пройденного материала дисциплин модуля «Информатика», дисциплин: «Языки программирования», «Основы программной инженерии» и «Конструирование компиляторов».

### **Семестр I**

Изучение стандартных алгоритмов, принципов построения алгоритмов для решения типовых задач, получение студентами базовых навыков программирования.

### **Семестр II**

Изучение архитектуры компьютера, а также обучение студентов написанию программ на языке ассемблера и дальнейшее развитие навыков программиста. Курс связан с решением задач на языке ассемблера.

### **Семестр V**

Изучение методов разработки программ с использованием виртуальных функций абстрактных классов и стандартной библиотеки шаблонов. Изучение методов разработки и применения статических и динамических библиотек. Ознакомиться с приемами компонентного программирования в языках C++, C# и Java Изучить приемы функционального программирования.

### **Семестр VI**

Ознакомление студентов с методами и технологиями программирования в среде .NET.

### **Семестр VII**

Целью и задачей изучения дисциплины является развитие логического мышления студентов, освоение студентами алгоритмов работы компиляторов, методов конструирования компиляторов.

### **Основные задачи дисциплины**

#### **Семестр I**

- на основе интуитивного понятия алгоритма выполнить формализацию этого понятия;
- рассмотреть основные объекты, которые можно назвать формализацией понятия алгоритм: машина Тьюринга и нормальные алгоритмы Маркова;
- изучить базовые вопросы, связанные со сложностью алгоритмов и алгоритмической разрешимостью;
- дать понятие формальных языках и способах их описания;
- изучить алгоритмические языки, на примере языка программирования Паскаль;
- изучить классические динамические структуры данных и основные алгоритмы для работы с ними;
- рассмотреть вопросы, связанные с методами разработки программных средств.

#### **Семестр II**

- изучить архитектуру компьютера, обучить студентов написанию программ на языке ассемблера.

**Семестр V**

– освоить приемы ООП, компонентного и функционального программирования.

**Семестр VI**

В течение курса решить набор задач по реализации доступа к данным, юнит-тестирования, логированию исключительных ситуаций, отображения данных (WebForms), также реализовать утилитное приложение для редактирования конфигурационного файла (Web.config, App.config).

В качестве дополнительного задания - реализация модуля инстанцирования объектов с использованием Reflection - IoC контейнер, реализующий Dependency Injection паттерн. При решении задач вводится понятие основных шаблонов проектирования и примеры их реализации.

**Семестр VII**

– развитие логического мышления студентов, освоение студентами алгоритмов работы компиляторов, методов конструирования компиляторов.

**2. Место дисциплины в структуре ООП**

Дисциплина «Практикум на ЭВМ» входит в общепрофессиональный блок базовой части ОС МГУ по направлению подготовки 010400.62 «Прикладная математика и информатика». «Практикум на ЭВМ» включает в себя как семинарские занятия, так и практическое выполнение заданий на ЭВМ.

Для успешного освоения дисциплины каждого последующего семестра студент должен обладать основами знаний по информатике и математике в рамках школьной программы и материалы практикумов предшествующих семестров, а также теоретический материал курсов, которые поддерживает практикум.

**3. Требования к результатам освоения дисциплины**

В результате освоения дисциплины обучающийся должен:

*Знать:*

**Семестр I**

- основные свойства алгоритмов;
- основные формализации алгоритмов – машину Тьюринга и Нормальные алгоритмы Маркова;
- о сложности стандартных алгоритмов;
- способы формального описания языков;
- синтаксис и семантику языка высокого уровня;
- способы разработки программного обеспечения;
- типовые динамические структуры данных;
- знать об основных управляющих структурах Паскаля.

**Семестр II**

- устройство ЭВМ на уровне прикладного программиста;
- понимать и объяснять тексты программ на языке ассемблера;
- основные команды языка Ассемблера и уметь их использовать;

**Семестр V**

- принципы положенные в основу ООП,
- принципы организации и использования стандартных библиотек и шаблонов
- принципы организации статических и динамических библиотек,
- принципы создания пользовательского интерфейса с использованием компонент ОС и стандартных библиотек графического интерфейса.

**Семестр VI**

- как создается корпоративное приложение;
- какова архитектура корпоративного приложения;
- какие основные модули и их предназначение на примере технологий .NET.

**Семестр VII**

- основные объекты курса «Конструирование компиляторов» - грамматики, регулярные множества и выражения, конечные автоматы, связь этих объектов.
- технологии динамической организации памяти, трансляции арифметических и логических выражений, генерации оптимального кода.

***Уметь:*****Семестр I**

- применять и адаптировать для решения задач стандартные алгоритмы;
- составлять машины Тьюринга и нормальные алгоритмы Маркова для решения задач;
- строить металингвистические формулы и синтаксические диаграммы для модельных языков;
- понимать и объяснять тексты программ на Паскале;
- создавать, использовать, модифицировать программы, работающие со стандартными структурами данных;
- пользоваться основными управляющими структурами Паскаля;
- создавать и применять процедуры и функции, в том числе и рекурсивные, в программах на Паскале;
- пользоваться процедурами и функциями, в том числе и рекурсивными, в программах на Паскале;
- создавать, использовать, модифицировать программы, работающие с динамическими структурами данных

**Семестр II**

- реализовывать процедуры, в том числе и рекурсивные, на языке ассемблера;
- создавать многомодульные программы;
- использовать макросредства;
- пользоваться конструкциями структурного программирования;
- разрабатывать программы с помощью пошаговой детализации;
- использовать псевдокод

**Семестр V**

- применять принципы ООП для разработки прикладных программ,
- пользоваться стандартными библиотеками шаблонов
- создавать и использовать статические и динамические библиотеки в языках – C++ и C#,
- создавать и использовать библиотеки в языке Java,

- разрабатывать прикладные программы с использованием графических компонент ОС.

### **Семестр VI**

- работать со всеми основными конструкциями языка СИ#, используя для работы VS Visual Studio 2010.

### **Семестр VII**

- строить грамматики и конечные автоматы, осуществлять эквивалентные преобразования грамматик.

### ***Владеть:***

- профессиональными знаниями теории алгоритмических систем и методов разработки и реализации программного обеспечения;
- обладать начальными практическими навыками программиста, т.е. уметь тестировать и отлаживать простые программы, пользоваться отладчиком.

### **Универсальные, профессиональные и специализированные компетенции, которыми должен обладать студент в результате освоения дисциплины**

#### **Универсальные компетенции:**

##### **а) общенаучные:**

владение методологией научных исследований в профессиональной области (ОНК-4);

##### **б) инструментальные:**

владение навыками использования программных средств и работы в компьютерных сетях, использования ресурсов Интернет; владение основными методами, способами и средствами получения, хранения, переработки информации (ИК-3);

способность использовать современную вычислительную технику и специализированное программное обеспечение в научно-исследовательской работе (ИК-4);

##### **в) системные:**

способность к самостоятельному обучению и разработке новых методов исследования, к изменению научного и научно-производственного профиля деятельности (СК-3);

#### **Профессиональные компетенции:**

в области научно-исследовательской деятельности:

способность демонстрации общенаучных базовых знаний естественных наук, прикладной математики и информатики, понимание основных фактов, концепций, принципов и теорий, связанных с прикладной математикой и информатикой (ПК-1);

способность понимать и применять в исследовательской и прикладной деятельности современный математический аппарат (ПК-2);

в проектной и производственно-технологической деятельности:

способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии (ПК-3);

способность осваивать информационные и суперкомпьютерные технологии при решении практических задач (ПК-4);

способность собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности

(ПК-5);

способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы (ПК-8);  
в инновационной деятельности:

способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии (ПК-10);

способность осуществлять целенаправленный поиск информации о технологических достижениях в сети Интернет и из других источников (ПК-11).

## 1. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### а) Общая трудоемкость дисциплины:

-12 зачётных единиц,  
- 432 академических часа. В том числе: 252 часа семинаров, 180 часов самостоятельной работы.

#### **Семестр I**

- 3 зачетные единицы,  
- 108 академических часов. В том числе: 72 часа семинаров, 36 часов самостоятельной работы

#### **Семестр II**

- 3 зачетные единицы,  
- 108 академических часов. В том числе: 72 часа семинаров, 36 часов самостоятельной работы

#### **Семестр V**

- 2 зачетные единицы,  
- 72 академических часов. В том числе: 36 часов семинаров, 36 часов самостоятельной работы

#### **Семестр VI**

- 2 зачетные единицы,  
- 72 академических часов. В том числе: 36 часов семинаров, 36 часов самостоятельной работы

#### **Семестр VII**

- 2 зачетные единицы,  
- 72 академических часов. В том числе: 36 часов семинаров, 36 часов самостоятельной работы

### б) Тематический план

Семинарские занятия проходят в формате занятий в аудитории с разбором и решением задач у доски и в формате занятий в компьютерных классах на ЭВМ.

Таблица 1.

### ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Название темы	Количество часов			Формы текущего контроля успеваемости (по темам) / Форма промежуточной аттестации (по семестрам)
		Л	С(П,Лб) ауд./на ЭВМ	СРС	
1	2	3	4	5	6
<b>Семестр I</b>					
<b>Раздел 1. Введение в алгоритмы</b>					
1	Системы счисления		2	1	Письменная самостоятельная работа (одна задача или пример)
2	Машины Тьюринга		2/2	2	Проверка работ на ЭВМ
3	Нормальные алгоритмы Маркова		2/2	2	Проверка работ на ЭВМ
4	Металингвистические формулы. Синтаксические диаграммы. <b>Контрольная работа</b> по разделу “Введение в алгоритмы” (2 часа)		4	2	Письменная контрольная работа.
<b>Раздел 2. Алгоритмический язык Паскаль</b>					
5	Введение в Паскаль		4/2	3	Проверка работ на ЭВМ
6	Сложные операторы. Нестандартные скалярные типы		18/2	5	Проверка работ на ЭВМ
7	Регулярные типы		6/4	5	Проверка работ на ЭВМ
8	Комбинированные типы данных		2	1	Письменная самостоятельная работа (одна задача или пример)
9	Процедуры и функции		8/4	6	Проверка работ на ЭВМ
10	Остальные производные типы Паскаля		6/2	4	Письменная самостоятельная работа (одна задача или пример). Проверка работ на ЭВМ
<b>Раздел 3. Динамические структуры данных</b>					
11	Ссылки, списки		2	1	Письменная самостоятельная работа (одна задача или пример)

1	2	3	4	5	6
12	Стеки, очереди		2	1	Письменная самостоятельная работа (одна задача или пример)
13	Деревья. Двоичные деревья поиска. AVL-деревья. <b>Итоговая контрольная работа (2 часа)</b>		4/2	3	Проверка работ на ЭВМ
<b>Всего, часов</b>			<b>72</b>	<b>36</b>	<b>Зачёт с оценкой</b>
<b>Семестр II</b>					
<b>Раздел 1. Задания для самостоятельной работы</b>					
1	Задание 1. Паскаль, вычисление площади фигуры, ограниченной тремя кривыми		2	4	Проверка работ на ЭВМ
2	Задание 2. Паскаль, динамические структуры данных		2	4	Проверка работ на ЭВМ
3	Задание 3. Ассемблер, обработка литерных данных		2	4	Проверка работ на ЭВМ
4	Задание 4. Ассемблер, динамические структуры данных		2	4	Письменная самостоятельная работа (одна задача или пример). Проверка работ на ЭВМ
<b>Раздел 2. Введение в архитектуру ЭВМ</b>					
5	Программирование на учебной 3-х адресной ЭВМ УМ-3		2	2	Письменная самостоятельная работа (одна задача или пример)
6	Машинное представление целых чисел на IBM PC		2	2	Письменная самостоятельная работа (одна задача или пример)
<b>Раздел 3. Программирование на языке Ассемблера</b>					
7	Введение в Ассемблер		2/2	2	Проверка работ на ЭВМ
8	Основные команды языка Ассемблера		8/6	8	Проверка работ на ЭВМ
9	Массивы. Программный стек. Сегментирование		6/6	6	Проверка работ на ЭВМ
10	Процедуры на языке Ассемблера		4/2	4	Проверка работ на ЭВМ
11	Динамические структуры данных		4/2	4	Проверка работ на ЭВМ
12	Макросредства языка Ассемблера		2/2	2	Проверка работ на ЭВМ
13	Многомодульные программы		2	2	Проверка работ на ЭВМ
14	<b>Контрольная работа</b>		4	4	Проверка контрольной работы
<b>Всего, часов</b>			<b>72</b>	<b>36</b>	<b>Зачёт с оценкой</b>

1		2	3	4	5
<b><u>Семестр V</u></b>					
1	Виртуальные функции и абстрактные классы		4	4	Проверка работ на ЭВМ
2	Стандартная библиотека шаблонов.		4	4	Проверка работ на ЭВМ
3	Создание и использование статических и динамических библиотек в C++ и C#		4	4	Проверка работ на ЭВМ
4	Создание и использование пакетов в языке Java. Интерфейсы		4	4	Проверка работ на ЭВМ
5	Использование пакетов для создания библиотек языка Java.		4	4	Проверка работ на ЭВМ
6	Компонентное программирование в C++.		4	4	Проверка работ на ЭВМ
7	Компонентное программирование в C#.		4	4	Проверка работ на ЭВМ
8	Компонентное программирование в Java.		4	4	Проверка работ на ЭВМ
9	Функциональное программирование в языке F#		4	4	Проверка работ на ЭВМ
<b>Всего, часов</b>			<b>36</b>	<b>36</b>	<b>Зачёт с оценкой</b>
<b><u>Семестр VI</u></b>					
<b>Раздел 1. Технологии программирования .NET</b>					
	DAL. Реализация с помощью ADO.NET		4	4	Проверка работ на ЭВМ
	Logging, unit-testing. Реализация модуля логирования. Написание юнит-тестов.		4	4	Проверка работ на ЭВМ
	ASP.NET. Реализация простейшего приложения.		8	8	Проверка работ на ЭВМ
	Windows Forms. Реализация утилиты.		4	4	Проверка работ на ЭВМ
<b>Раздел 2. Выполнение задания “Форум”</b>					
	Задание “Форум”		16	16	Проверка работ на ЭВМ
<b>Всего, часов</b>			<b>36</b>	<b>36</b>	<b>Зачёт с оценкой</b>
<b><u>Семестр VII</u></b>					
1	Граматики (определение грамматики, классификация и примеры)		2	2	
2	Регулярные множества и выражения, конечные автоматы (PM, PB, KA).		2	2	
3	Построение ДКА по НКА. Алгоритм построения КА по PB.		2	2	
4	Связь PM, KA и регулярных грамматик.		4	2	

5	<b>Контрольная работа № 1.</b>		1	4	Проверка письменной работы. Консультация.
6	Автоматы с МП (определения и примеры). Связь МП-автоматов и КС-грамматик. Приведённые грамматики (алгоритмы)		4	4	
7	Критерий бесконечности (применение его при доказательстве не-КС-грамматик) и НС-грамматики (алгоритм построения). Функции FIRST и FOLLOW. LL(k)-грамматики (определения и примеры).		4	4	
8	Необходимые и достаточные условия. Построение LL(1)-анализатора.		4	4	
9	LR(k)-грамматики (определения и примеры).		2	2	
10	LR(k)-анализатор. Функция EFF.		2	2	
11	Необходимые и достаточные условия LR(k). Атрибутные грамматики.		4	2	
12	<b>Контрольная работа № 2.</b>		1	4	Проверка письменной работы. Консультация.
13	Генерация кода		4	2	
<b>Всего, часов</b>			<b>36</b>	<b>36</b>	<b>Зачёт с оценкой</b>

где: Л – лекции, С – семинарские занятия, П – практические занятия, Лб – лабораторные занятия, СРС – самостоятельная работа студентов.

## 2. СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ

### а) Планы семинарских занятий

Семинарские занятия проходят в формате занятий в аудитории с разбором и решением задач у доски и в формате занятий в компьютерных классах на ЭВМ.

#### Семестр I

**Семинар 1.** Системы счисления.

**Семинар 2.** Машины Тьюринга.

**Семинар 3.**

1. Нормальные алгоритмы Маркова.
2. Работа на ЭВМ: Нормальные алгоритмы Маркова.

**Семинар 4.** Металингвистические формулы. Синтаксические диаграммы.

**Семинар 5.**

1. Введение в Паскаль.
2. Работа на ЭВМ: Введение в Паскаль

**Семинар 6.**

1. Сложные операторы. Нестандартные скалярные типы.
2. Работа на ЭВМ: Сложные операторы. Нестандартные скалярные типы.

**Семинар 7.**

1. Регулярные типы.
2. Работа на ЭВМ: Регулярные типы.

**Семинар 8.** Комбинированные типы данных.

**Семинар 9.**

1. Процедуры и функции.
2. Работа на ЭВМ: Процедуры и функции.

**Семинар 10.**

1. Остальные производные типы Паскаля.
2. Работа на ЭВМ: Остальные производные типы Паскаля.

**Семинар 11.** Ссылки, списки.

**Семинар 12.** Стеки, очереди.

**Семинар 13.**

1. Деревья. Двоичные деревья поиска. AVL-деревья.
2. Работа на ЭВМ: Деревья. Двоичные деревья поиска. AVL-деревья.

**Семинар 14**

1. Резерв.
2. Контрольная работа

#### Семестр II

Из 4-х учебных часов занятия по практикуму в неделю необходимо 3 часа отводить для семинарских занятий и один час – на сдачу результатов выполнения практических работ в терминальном классе.

**Семинар 1.**

1. Выдача задания N1 (Паскаль, вычисление площади фигуры, ограниченной тремя кривыми) (см. Примечание 1).
2. Выдача задания N2 (Паскаль, динамические структуры данных, обработка слов) (см. Примечание 1).

**Семинар 2.**

1. Программирование на учебной 3-х адресной ЭВМ (УМ-3). См. “Методические указания”.
2. Работа на ЭВМ

**Семинар 3.**

1. Машинное представление целых чисел на IBM PC (прямой, обратный и дополнительный код), сложение и вычитание, правила установки флагов CF, OF, ZF и SF.
2. Работа на ЭВМ.

**Семинар 4.**

1. Язык Ассемблера. Директивы определения данных (DB, DW, DD), команды пересылок. Оператор PTR, директива EQU.
2. Работа на ЭВМ.

**Семинар 5.**

1. Арифметические команды над целыми числами. Программирование формул, арифметика “длинных” целых чисел.
2. Безусловные и условные переходы. Работа на ЭВМ.

**Семинар 6.**

1. Команда сравнения Циклы. Обработка символов.
2. Ввод и вывод с помощью макрокоманд INCH, ININT, FLUSH, OUTCH, OUTINT, OUTSTR и др. Макрокоманда FINISH. Работа на ЭВМ.

**Семинар 7.**

1. Массивы: описание, индексирование. Команда загрузки адреса LEA. Задачи на обработку массивов. Операторы TYPE, LENGTH и SIZE.
2. Структуры: описание, смысл имени поля. Операторы TYPE, LENGTH и SIZE применительно к массивам структур. Работа на ЭВМ.

**Семинар 8.**

1. Сегментные регистры, соглашение по умолчанию. Простейшие полные программы.
2. Контрольная работа (порядка 3-х задач по пройденным темам). Работа на ЭВМ.

**Семинар 9.**

1. Работа со стеком. Процедуры, команды CALL и RET. Способы передачи параметров через регистры.
2. Правила работы на Языке Ассемблера IBM PC: редактирование, компиляция, отладка. Выдача задания N3 (Ассемблер, задание N6 из “Практикума работы на ЭВМ”). Работа на ЭВМ.

**Семинар 10.**

1. Процедуры: стандартные соглашения о связях, передача параметров через стек.
2. Программирование процедур и функций. Работа на ЭВМ.

**Семинар 11.**

1. Логические данные и операции над ними. Задачи на вычисление логических выражений, умножение и деление на степень 2.
2. Множества, операции над ними, распаковка и упаковка данных. Записи, операторы WIDTH, MASK, смысл имени поля. Работа на ЭВМ.

**Семинар 12.**

1. Команды обработки строк, префиксы повторения, флаг DF.
2. Макрокоманды NEW и DISPOSE Работа с динамическими структурами данных (это подготовка к выполнению задания на ЭВМ, на лекциях этой темы нет). Работа на ЭВМ.

**Семинар 13.**

1. Макросредства Ассемблера. Простейшие макроопределения. Директива LOCAL. Выдача задания N4 (Ассемблер, задание N7 из “Практикума работы на ЭВМ”) (см. Примечание 2).

2. Условное ассемблирование. Операторы отношения (EQ, LT и т.д.), блоки IF, директива EXITM. Решение задач на макросредства. Работа на ЭВМ.

#### **Семинар 14.**

1. Простейшие двухмодульные программы. Директивы PUBLIC и EXTRN.
2. Работа на ЭВМ.

#### **Семинар 15.**

1. Резерв.
2. Контрольная работа.

### **Семестр V**

Практические занятия подразумевают работу студентов на ЭВМ в пакете VS Visual 2008 и Java(TM) Se

**Семинар 1.** Виртуальные функции и абстрактные классы.

**Семинар 2.** Стандартная библиотека шаблонов.

**Семинар 3.** Создание и использование статических и динамических библиотек в C++ и C#.

**Семинар 4.** Создание и использование пакетов в языке Java. Интерфейсы

**Семинар 5.** Использование пакетов для создания библиотек языка Java

**Семинар 6.** Компонентное программирование в C++.

**Семинар 7.** Компонентное программирование в C#.

**Семинар 8.** Компонентное программирование в Java.

**Семинар 9.** Функциональное программирование F#.

### **Семестр VI**

Практические занятия подразумевают работу студентов на ЭВМ в пакете VS Visual Studio 2010, на языке СИ#.

#### **Семинар 1.**

1. DAL. Реализация с помощью ADO.NET
2. Работа на ЭВМ.

#### **Семинар 2.**

1. Logging, unit-testing. Реализация модуля логирования. Написание юнит-тестов.
2. Работа на ЭВМ.

#### **Семинар 3.**

1. ASP.NET. реализация простейшего приложения.
2. Работа на ЭВМ.

#### **Семинар 4.**

1. Windows Forms. Реализация утилиты.
2. Работа на ЭВМ.

#### **Семинар 5.**

1. Выполнение задания “Форум”.
2. Работа на ЭВМ.

## **Семестр VII**

**Семинар 1.** Грамматики (определение грамматики, классификация и примеры)

**Семинар 2.** Регулярные множества и выражения, конечные автоматы (PM, PB, KA).

**Семинар 3.** Построение ДКА по НКА. Алгоритм построения КА по PB.

**Семинар 4.** Связь PM, KA и регулярных грамматик.

### **Контрольная работа № 1.**

**Семинар 5.** Автоматы с МП (определения и примеры). Связь МП-автоматов и КС-грамматик. Приведённые грамматики (алгоритмы)

**Семинар 6.** Критерий бесконечности (применение его при доказательстве не-КС-грамматик) и НС-грамматики (алгоритм построения).

Функции FIRST и FOLLOW. LL(k) –грамматики (определения и примеры).

**Семинар 7.** Необходимые и достаточные условия. Построение LL(1)-анализатора.

**Семинар 8.** LR(k)-грамматики (определения и примеры).

**Семинар 9.** LR(k)-анализатор. Функция EFF.

**Семинар 10.** Необходимые и достаточные условия LR(k). Атрибутные грамматики.

### **Контрольная работа № 2.**

**Семинар 11.** Генерация кода

Модель машины

Динамическая организация памяти

Трансляция арифметических выражений

Трансляция логических выражений

Генерация оптимального кода методами синтаксического анализа

### 3. РЕКОМЕНДУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Работа в аудитории: решение и разбор задач у доски; консультации перед экзаменом.

Работа в компьютерном классе: решение задач на ЭВМ; самостоятельная подготовка к занятиям с использованием Интернет-ресурсов и электронных пособий кафедры.

Внеаудиторная работа: изучение пройденных на лекциях тем, самостоятельное изучение литературы по дисциплине.

#### 4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

##### а) Оценочные средства для текущего контроля успеваемости

##### Семестр I

- в качестве текущего контроля проводится несколько самостоятельных работ по основным темам и две контрольные работы.

##### **Пример контрольной работы по разделу “Введение в алгоритмы” (2 часа)**

1. а) Сформулируйте тезис Тьюринга. б) Объясните, почему его нельзя доказать. в) Приведите доводы в пользу этого тезиса.

2. В алфавите  $A=\{a,b,c\}$  заданы алгоритмы  $P$ ,  $Q$  и  $R$ :

$P: ca \rightarrow ac$                        $Q: cb \rightarrow bc$                        $R: \begin{matrix} cb \rightarrow bc \\ ca \rightarrow ac \end{matrix}$

Является ли  $R$  композицией  $Q$  и  $P$ , т. е. Верно ли что  $R=Q \circ P$  или  $R(\omega)=P(Q(\omega))$ , где  $\omega$  – любое слово из  $A^*$ ? Дать ответ (да/нет) и обосновать его.

3. Определить, является ли самоприменимым следующий алгоритм  $W$  в расширенном алфавите  $A=\{a, b, c\}$ . Дать ответ (да/нет) и привести обоснование.

$W: \begin{matrix} *a \rightarrow a^* \\ b \rightarrow bc \\ c \rightarrow cc \\ \rightarrow * \end{matrix}$

4. Для заданного нормального алгоритма  $W$  в расширенном алфавите  $A=\{a, b\}$  привести эквивалентный ему нормальный алгоритм  $R$  в алфавите  $A$ , состоящий из одного правила подстановки.

$W: \begin{matrix} +a \rightarrow a+ \\ +b \rightarrow b+ \\ + \rightarrow * \\ a^* \rightarrow *a \\ b^* \rightarrow *b \\ * \rightarrow a \\ \rightarrow + \end{matrix}$

5. Даны пять нормальных алгоритмов в расширенном алфавите  $A=\{a,b\}$ . Есть ли среди них алгоритмы, являющиеся композицией двух других алгоритмов из заданных? Если “да”, то указать все такие композиции. Если “нет” – ответ обосновать.

$U: \begin{matrix} ab \rightarrow \end{matrix}$                        $T: \begin{matrix} b \rightarrow a \end{matrix}$                        $W: \begin{matrix} ab \rightarrow \\ b \rightarrow a \end{matrix}$                        $Q: \begin{matrix} ab \rightarrow \\ b \rightarrow a \\ \rightarrow * \\ * \rightarrow \end{matrix}$                        $R: \begin{matrix} *b \rightarrow a^* \\ *a \rightarrow a^* \\ * \rightarrow \\ ab \rightarrow \\ \rightarrow * \end{matrix}$

6. Дан алфавит  $A=\{a, b\}$ . Приведите пример нормального алгоритма в расширенном алфавите  $A$ , который применим к любому слову из  $A^*$ , содержит не более трех правил подстановки и несамоприменим.

7. При помощи металингвистических формул описать состав вида:

Э Э С ... С К ... К О ... О Р В ... В

где Э – электровагоны, С – спальные вагоны (их количество  $i>0$ ), О – общие (их количество  $j>0$ ), К – купейные (их количество  $k \geq i+j$ , т. е. столько же, сколько всего спальных и общих вагонов или больше), Р – ресторан, В – багажные (их количество  $3 \geq n \geq 0$ ).

8. Дано определение понятия список:

$\langle \text{список} \rangle ::= \langle \text{атом} \rangle \mid \langle \text{список} \rangle, \langle \text{атом} \rangle \mid (\langle \text{список} \rangle) \mid \langle \text{пусто} \rangle$

$\langle \text{атом} \rangle ::= X \mid Y$

$\langle \text{пусто} \rangle ::=$

- 8.1) записать понятие <список> в виде синтаксической диаграммы;  
 8.2) указать, какие из приведенных цепочек не являются списками и почему:  
 а) (X,)    б) (X Y)    в) (, Y)    г) X,,Y    д) ((,X),Y),X    е) ((X),())
9. Осуществить перевод из одной системы счисления в другую:  $F1E35, A09_{16}$   
 $= ?_8$

### Пример самостоятельной работы “Циклы и стандартные типы” (1 час)

- Указать, сколько раз будет выполняться тело следующего оператора цикла:  
`k:= ord(chr(ord('Z')+3))- ord('Z');`    Ответ:  
`for i:=1 to k+3 do k:=k+1;`
- Для каждого из приведенных выражений на Паскале указать его значение или слово "ошибка", если выражение ошибочно.    Ответы:  
 а) `trunc(-1.8) = -trunc(1.8)`    а)  
 б) `20 div 4/2`    б)  
 в) `pred(ord(chr(0)))`    в)
- На плоскости (x,y) нарисовать область, в которой и только в которой истинно выражение:  
`(trunc(y)=0) and (round(x)=0)`  
 Дано 50 вещественных чисел. Определить, образуют ли они возрастающую последовательность. В случае положительного ответа напечатать "Да", в случае отрицательного ответа - "Нет".

### Пример самостоятельной работы “Массивы” (1 час)

- Имеются описания типов:  
`type towns=(Moscow, London, Paris, Rome);`  
`months=(january, february, march, april, may, june, jule, august, september, october, november, december);`  
`tempdata=array[towns,months] of real;`  
 Массивы типа tempdata используются для хранения среднемесячных температур в городах. Описать процедуру minhottown с параметром t типа tempdata, которая по данным из массива t определяет в каждом городе температуру самого жаркого месяца и распечатывает название города, в котором эта температура является наименьшей по сравнению с другими городами.
- Указать типы переменных a, b и c, при которых данный оператор не содержит ошибки, или указать причину, по которой оператор содержит ошибку независимо от типов переменных, используемых в нем.  
`a[b] := chr(ord(c)) = succ(b)`
- Нарисовать на координатной плоскости (x,y) область в которой истинно данное выражение:  
`(abs(y)<ord(true)) and odd(trunc(x))`

### Пример самостоятельной работы “Записи, процедуры и множества” (1 час)

- Описать процедуру Хвостисты(m), которая по данным об успеваемости студентов, переданным в m – параметре-массиве типа course, печатает имена студентов, имеющих хоть одну двойку.  
`Type student = record Name: packed array [1..20] of char;`  
`Marks: array [1..5] of 2..5`  
`end;`  
`group = array [1..25] of student;`  
`course = array [1..16] of group;`
- Определить, что будет выдано на печать:  
`program Pr(output);`

```
var a,b,c,d: integer;
procedure P(x:integer; var a:integer);
var c:integer;
begin x:=5; a:=6; c:=7; d:=8; writeln( x,a,c,d) end;
begin a:=1; b:=2; c:=3; d:=4; P(a,b); writeln(a,b,c,d) end.
```

3. function F(n:integer):integer;  
begin if n>100 then f:=n-10 else f:=f(f(n+11)) end;  
Вычислить F(105) и F(80).

4. var R2: record A,B,C: integer end;  
R3: record A,D: integer;  
B: record C,E: integer end end;  
with R3,B,R2 do begin A:=1; B:=2; C:=3; D:=4; E:=5 end  
Для заданного оператора присоединения написать эквивалентный ему составной оператор.

5. Найти все ошибки в следующем фрагменте программы:  
Type M=set of char;  
function f(a,b:M; x:char):M;  
begin if a\*b=0 then a:=[x] else if a<b then a:=b+x else  
if ord(x) in a-b then a:=a-[x..'<=']; f:=a+b end;

### Пример самостоятельной работы “Динамические структуры” (1 час)

1. Привести описание рекурсивной функции, вычисляющей в однонаправленном списке элементов целого типа число элементов, значение которых равно заданному целому числу.
2. Привести описание процедуры, которая преобразует любой заданный ей однонаправленный список элементов целого типа, удаляя в нем N-ный по порядку элемент списка. Если число элементов в списке меньше N, то список не изменяется.
3. Привести описание функции, подсчитывающей в любом заданном текстовом файле число непустых строк, начинающихся с символа пробела.

### Семестр II

- в качестве текущего контроля проводится несколько самостоятельных работ по основным темам, а также выполняются 4 практических задания в течение семестра с последующей проверкой на ЭВМ.

**Пример 1:** вычислить для целых X и Y:

$$Y = \begin{cases} X+2, & \text{при } X < 2 \\ 2, & \text{при } X = 2 \\ 2*(X+2), & \text{при } X > 2 \end{cases}$$

**Пример 2:** суммирования элементов массива с переадресацией команд (рекомендуется далее использовать на занятиях мнемонические обозначения кодов операций):

$$S = \sum_{i=1}^{100} x[i]$$

### Семестр V

- на каждом занятии проверяется домашнее задание на ЭВМ.

**Семестр VI**

- домашние задания по решению типовых задач с последующей проверкой на ЭВМ. Задание «Форум».

**Семестр VII**

- в качестве текущего контроля проводятся две контрольные работы.

**Образцы контрольных работ****Контрольная работа № 1.**

- А. Построение грамматики (КС, НС и т.п.) по языку с доказательством (по индукции и т.п.)
- Б. Определить: является ли заданное множество регулярным.
- В. Построить КА, принимающего заданный язык.
- Г. Построение автомата по РВ.
- Д. Построение по РМ конечного автомата (КА) и регулярных грамматик
- Е. Является ли заданный язык КС-языком.

**Контрольная работа № 2.**

- А. Построение МП-автоматов.
- Б. Построение КС-грамматики по МП-автомату и обратно.
- В. Эквивалентные преобразования КС-грамматик.
- Г. Проверка условия LL(k).
- Д. Построение LL(1)-анализатора и демонстрация его работы для конкретного слова.
- Е. Проверка условия LR(k).
- Ж. Построение LR(k)-анализатора и демонстрация его работы для конкретного слова.
- З. Построение атрибутной грамматики.

**Самостоятельная работа по изучению данной дисциплины включает:**

- проработку теоретических основ материала;
- изучение отдельных тем или вопросов, которые предусмотрены для самостоятельной работы;
- подготовку к практическим занятиям на ЭВМ;
- систематизацию изученного материала по курсу;

**Семестр I**

Студентам выдаются задачи для решения в качестве домашнего задания с последующей проверкой в аудитории. Задачи выдаются из учебного пособия [3]. Примеры задач также даны в приложении. Курс не подразумевает выполнение реферативных работ.

**Задание 1. Паскаль, вычисление площади фигуры, ограниченной тремя кривыми**

С заданной точностью  $\epsilon_{ps}$  вычислить площадь плоской фигуры, ограниченной тремя кривыми, уравнения которых  $y=f_1(x)$ ,  $y=f_2(x)$  и  $y=f_3(x)$  определяются вариантом задания.

При решении задачи необходимо:

- с некоторой точностью  $\epsilon_{ps1}$  вычислить абсциссы точек пересечения кривых, используя предусмотренный вариантом задания метод приближенного решения уравнения  $F(x)=0$ ; отрезки, где программа будет искать точки пересечения и где применим используемый метод, определить вручную;

- представить площадь заданной фигуры как алгебраическую сумму определенных интегралов и вычислить эти интегралы с некоторой точностью  $\epsilon_2$  по квадратурной формуле, предусмотренной вариантом задания.

Величины  $\epsilon_1$  и  $\epsilon_2$  подобрать вручную так, чтобы гарантировалось вычисление площади фигуры с точностью  $\epsilon$ .

Во всех вариантах  $\epsilon = 0.001$ .

Уравнения кривых  $y = f_i(x)$ :

- 1)  $f_1 = 2x + 1$        $f_2 = x^5$     $f_3 = (1-x)/3$
- 2)  $f_1 = 3(0.5/(x+1) + 1)$     $f_2 = 2.5x - 9.5$     $f_3 = 5/x$  ( $x > 0$ )
- 3)  $f_1 = \exp(-x) + 3$     $f_2 = 2x - 2$        $f_3 = 1/x$
- 4)  $f_1 = \exp(x) + 2$     $f_2 = -1/x$        $f_3 = -2(x+1)/3$
- 5)  $f_1 = 0.35x^2 - 0.95x + 2.7$     $f_2 = 3x + 1$        $f_3 = 1/(x+2)$
- 6)  $f_1 = 0.6x + 3$        $f_2 = (x-2)^3 - 1$     $f_3 = 3/x$
- 7)  $f_1 = \ln(x)$        $f_2 = -2x + 14$        $f_3 = 1/(2-x) + 6$
- 8)  $f_1 = \exp(x) + 2$     $f_2 = -2x + 8$        $f_3 = -5/x$
- 9)  $f_1 = 3/((x-1)^2 + 1)$        $f_2 = \sqrt{x+0.5}$     $f_3 = \exp(-x)$
- 10)  $f_1 = 1 + 4/(x^2 + 1)$     $f_2 = x^3$     $f_3 = 2 - x$

## Задание 2. Паскаль, динамические структуры данных

Дана непустая последовательность слов, в каждом из которых содержится от 1 до 6 заглавных латинских букв; соседние слова разделены запятой, за последним словом следует точка. Требуется ввести эту последовательность слов в память ЭВМ, преобразовав ее во внутреннее представление (см. ниже), а затем распечатать в алфавитном порядке:

- 1) все слова последовательности;
- 2) все слова с указанием для каждого из них его порядкового номера в исходной последовательности;
- 3) все различные слова с указанием для каждого из них числа его вхождений в исходную последовательность;
- 4) все различные слова с указанием для каждого из них порядкового номера его первого вхождения в исходную последовательность;
- 5) сначала все однобуквенные слова, затем все двухбуквенные слова и т.д.;
- 6) сначала все однобуквенные слова с указанием для каждого из них его порядкового номера в исходной последовательности, затем аналогичным образом все двухбуквенные слова и т.д.;
- 7) сначала все различные однобуквенные слова с указанием для каждого из них числа его вхождений в исходную последовательность, затем аналогичным образом все различные двухбуквенные слова и т.д.;
- 8) сначала все различные однобуквенные слова с указанием для каждого из них порядкового номера его первого вхождения в исходную последовательность, затем аналогичным образом все различные двухбуквенные слова и т.д.;
- 9) та же задача, что и в варианте 1;
- 10) та же задача, что и в варианте 3;
- 11) та же задача, что и в варианте 4.

В качестве внутреннего представления последовательности слов использовать:

в вариантах 1-4 - однонаправленный список из слов, упорядоченных по алфавиту;

в вариантах 5-8 - массив из 6 списков, в  $k$ -ом из которых хранятся  $k$ -буквенные слова, упорядоченные по алфавиту;

в вариантах 9-11 - двоичное дерево поиска (в нем слева от каждой вершины-слова располагаются только те слова, что предшествуют ему по алфавиту, а справа - следующие за ним по алфавиту).

## **Семестр II**

Студентам выдаются задачи для решения в качестве домашнего задания с последующей проверкой в аудитории. Примеры задач даны в приложении. Курс также предусматривает выполнение четырех практических заданий:

### **Задание 3. Ассемблер, обработка литерных данных**

#### **ПОСТАНОВКА ЗАДАЧИ.**

Дан непустой текст (последовательность литер), содержащий не более 100 элементов, с точкой в качестве признака конца текста.

Требуется:

- ввести с клавиатуры данный текст и записать его в память ЭВМ;
- определить, обладает ли этот текст заданным свойством (свойство определяется вариантом задания);
- преобразовать текст по правилу 1, если он обладает заданным свойством, и по правилу 2 в противном случае (правила преобразования определяются вариантом задания);
- вывести на экран дисплея: исходный текст, номер примененного правила, преобразованный текст.

Свойство исходного текста:

- 1) Текст оканчивается заглавной латинской буквой, которая больше не встречается в тексте.
- 2) Текст начинается цифрой и оканчивается цифрой, причем эти цифры различны.
- 3) Текст начинается латинской буквой и оканчивается латинской буквой.
- 4) Текст содержит не менее трех латинских букв.
- 5) Текст содержит равное количество заглавных и строчных латинских букв.
- 6) Текст не содержит иных литер, кроме цифр и латинских букв

### **Задание 4. Ассемблер, динамические структуры данных**

#### **ПОСТАНОВКА ЗАДАЧИ**

Дана последовательность от 1 до 20 слов, каждое из которых содержит от 1 до 8 заглавных латинских букв; соседние слова разделены запятой, за последним словом следует точка.

Требуется ввести эту последовательность и преобразовать ее во внутреннее представление, а затем напечатать по алфавиту определенные слова с дополнительной информацией о каждом из них.

Внутреннее представление последовательности слов:

- 1) Список слов, упорядоченных по алфавиту.
- 2) Массив списков: список из однобуквенных слов, список из двухбуквенных слов и т.д. (в каждом из них слова упорядочены по алфавиту).
- 3) Двоичное дерево поиска (в нем слева от каждой вершины-слова должны находиться только те слова, что предшествуют этому слову по алфавиту, а справа - следующие за ним по алфавиту).

## **Семестр V**

Студентам выдаются методические указания для выполнения практических занятий и задание для выполнения 4-х самостоятельных работ.

**Задание для выполнения самостоятельной работы.** Задан формальный язык. Необходимо:

- описать формальную грамматику описывающую этот формальный язык;
- описать граф детерминированного автомата описывающего этот язык;

- описать алгоритм решения задачи распознавания слов языка из потока символов поступающих с клавиатуры и выдачи на печать распознанных слов заданного языка;
- разработать и отладить программу реализующую распознавание слов заданного языка в форме
  - консольного приложения в языке C++ или C#;
  - программы в языке C++, использующей средства Windows Form;
  - программы в языке C#, использующей средства Windows Form;
  - программы в языке Java, использующей библиотеку графических компонент Swing.

### **Вариант задания**

#### **описание языка $L_0$ .**

Язык включает в себя цепочки вида  $(ab(c^k))^n$  или  $ab^n$ , где  $k, n=1,2,3,\dots$  из алфавита  $\{a,b,c,/n\}$ .

### **Семестр VI**

Студентам выдаются задачи для решения в качестве домашнего задания с последующей проверкой в аудитории.

Также студенты получают следующее общее задание:

Оформление по правилам: Баула В.Г., Мещеряков Д.К. Учебное пособие по написанию программ при выполнении работ практикума на ЭВМ.

1. Выбрать сущность из любой предметной области. Например, Person с атрибутами:

Id, FirstName, LastName, Age, Gender. Реализовать хранение этой сущности в базе данных. Для чего создать соответствующую таблицу, объявить соответствующий класс, реализовать класс-manager, предоставляющий методы для сохранения сущности, ее удаления, получения сущности по идентификатору, получения списка сущностей. Использовать ADO.NET.

2. Реализовать модуль логирования возникающих исключительных ситуаций. Модуль будет представлять из себя класс. Конструктор класса будет иметь один параметр типа string, означающий путь к файлу, куда будет записываться информация об ошибке. У класса будет два перегруженных метода Log. Первый метод берет на вход параметр типа string. Второй метод - параметр типа Exception. Формат записи сообщения об ошибке: <дата записи>. <Сообщение об ошибке>. [<Stack\_trace>] Stack\_trace только в случае если вызывается метод с параметром типа Exception. Примечание: Реализованный модуль можно использовать в модуле доступа к данным в задании 1.

3. Реализовать юнит-тесты для всех методов модуля доступа к данным задания 1 с использованием NUnit или MbUnit.

4. Веб приложение (ASP.NET). Веб приложение представляет из себя три страницы. Первая страница (по умолчанию) - страница входа зарегистрированного пользователя в систему: два текстовых поля для ввода логина и пароля, кнопка с текстом "Войти", гиперссылка на страницу регистрации с текстом "Регистрация". При неправильном вводе логина/пароля выдавать соответствующее сообщение. Вторая страница - страница регистрации нового пользователя: три текстовых поля для ввода логина, пароля и подтверждения пароля, кнопка "Зарегистрировать". При нажатии кнопки "Зарегистрировать" проверять существует ли пользователь с введенным логином, если существует, выдавать соответствующее сообщение. Также проверять на совпадение введенного пароля и подтверждение пароля. Третья страница - страница со списком всех зарегистрированных пользователей. На эту страницу

будет перенаправляться удачно вошедший в систему пользователь. Примечание: в качестве слоя доступа к данным можно использовать модуль, написанный в задании 1.

5. Реализовать WinForm приложение для редактирования конфигурационного файла. Конфигурационный файл: web.config или app.config. Для простоты необходимо будет реализовать только редактирование секции <appSettings/> с фиксированным набором значений количеством от 5 штук (произвольно на выбор студента). Конфигурационный файл представляет из себя XML файл. Приложение будет состоять из одного окна с набором текстовых полей равных кол-ву значений в секции <appSettings/> и кнопки "Сохранить", по нажатию которой в секцию <appSettings/> конфигурационного файла будут записаны соответствующие значения из текстовых полей. При реализации данного приложения применить шаблон проектирования MVC или MVP.

## Семестр VII

Задания из банка задач факультета ВМК МГУ (обновляются, присылаются ежегодно)

### **б) Оценочные средства промежуточной аттестации**

К зачету допускаются студенты, отработавшие все текущие задания семестра.

## Семестр I

### **Пример зачётной контрольной работы (2 часа)**

1. Написать операторы, решающие задачу:

```
type единица = (миллиметр, сантиметр, дециметр, метр, километр);
длина = real;
var x: длина; p: единица;
```

Значение  $x$ , означающее длину в единицах  $p$ , заменить величиной той же длины в метрах.

2. for  $i := E1$  downto  $E2$  do  $S$  {  $i$  – переменная;  $E1, E2$  – какие-то выражения;  $S$  – оператор} Реализовать эквивалентный цикл при помощи операторов repeat и if (операторы for, while, goto не использовать).

3. Нарисовать на плоскости область, в которой и только в которой истинно выражение  $\text{succ}(\text{round}(\text{abs}(x)) > 2) = (\text{abs}(\text{trunc}(y)) < 3)$

4. Имеются описания типов  $\text{type matrix} = \text{array}(1..n, 1..n)$  of real;  $n$  – натуральная константа. Требуется описать процедуру или функцию, которая для заданной матрицы типа matrix выводит номера столбцов, для которых выполняется условие – все элементы столбца являются степенями 2 (т.е. вида  $2^k, k \geq 0$ ).

5. Указать типы переменных  $a, b$  и  $c$ , при которых данный оператор не содержит ошибки, или указать причину, по которой оператор содержит ошибку независимо от типов переменных, используемых в нем.

```
a[b] := odd(ord(b)+ord(a[chr(c)]))
```

6. var R2: record A,B,C: integer end;

```
R3: record A,D: integer;
```

```
B: record C,E: integer end end;
```

```
with R3,B,R2 do begin A:=1; B:=2; C:=3; D:=4; E:=5 end;
```

Для заданного оператора присоединения написать эквивалентный ему составной оператор.

7. Построить синтаксическую диаграмму понятия <определение перечислимого типа>. Можно использовать, считая заданными, понятия: <идентификатор>, <константа>, <служебное слово>.

8. Программа. На вход дан текст, составленный из малых латинских букв и скобок: '(', ')'. В конце текста точка (сама точка в текст не входит). Определить соблюдается ли в тексте баланс скобок и вывести «Текст сбалансирован», если соблюдается, и «Текст не сбалансирован» иначе.

9. Найти и объяснить все ошибки:

```
const n=50;
type  slovo = packed array [0..n-1] of letter;
      letter = 'a'..'z';
      massiv = array [letter] of real;
      vector = array [real] of integer;
      digits = array [false..true] of (0,1,2,3,4,5,6,7,8,9);
var    k: 1..maxint;
      x: array [1..k] of char;
      y: packed array [-n..n] of 0..0;
      z: array [(a,b,c)] of boolean;
```

10. Определить, что будет выдано на печать:

```
program A(output);
var a,b,c,d: integer;
procedure P(b:integer; var a:integer);
var c:integer;
begin  b:=5+b;
      a:=d+a;
      c:=a+b;
      d:=d-a;
      writeln( a,'-',b, '-',c, '-',d)
end;
begin
a:=10;
b:=20;
c:=30;
d:=40;
P(a,b);
writeln(a, '-',b, '-',c, '-',d)
end.
```

## Семестр II

### Пример зачётной контрольной работы (4 часа)

1. N EQU 10  
X DB N DUP (N DUP (?)) ; матрица X размером N x N  
Выписать фрагмент программы, который записывает в AL 1, если X – единичная матрица, и 0 – во всех остальных случаях.

2. X DW L

Указать для каждого приведённого ниже фрагмента вид команды JMP (то есть, является ли переход коротким или длинным, прямым или косвенным):

a) JMP L	б) LEA AX, CS:L	в) LEA BX, X	г) L: INC DX
MOV AX, BX	JMP AX	JMP WORD PTR [BX]	DEC DX
L: INC DX	L: MOV X, 0	L: INC BX	JMP L

Ответы: а) \_\_\_\_\_ б) \_\_\_\_\_  
в) \_\_\_\_\_ г) \_\_\_\_\_

3. Написать фрагмент программы, который при помощи команды OUTCH выводит беззнаковое число, записанное в AX в пятеричной системе счисления.

4. N EQU 50

X DD N DUP (?); массив чисел  $X_i$  ( $i=0\dots N-1$ )

Выписать фрагмент программы, который осуществляет сдвиг массива на две позиции вниз (то есть, после преобразования массива в  $X_i$  будет исходное значение  $X_{i-2}$  для  $i=1 \dots N-1$ , а в  $X_0$  – исходное значение  $X_{N-2}$ , в  $X_1$  –  $X_{N-1}$ ).

5. Пусть DS=1100h, SS=0FF00h, BX=8000h, BP=8001h, SI=9003h определить физический адрес первого операнда в следующих командах:

а) MOV [BX][SI-1], CX      б) SUB [BP-2], DH      в) ADD [SI+BP], AX

Ответы: а) \_\_\_\_\_ б) \_\_\_\_\_ в) \_\_\_\_\_

<p>6. A=10 B=A C EQU A D EQU +A X DB A, B, C, D A=2*A Y DB A, B, C, D</p>	<p>Указать начальные значения элементов массивов X и Y. Ответы: X: _____ Y: _____</p>
---	---

7. Описать структурный тип для представления времени в виде: часы, минуты, секунды; и две переменные T1 и T2 этого типа. Выписать фрагмент программы, в котором переменной T2 присваивается время, предшествующее T1 на 1 минуту (учесть смену суток).

8. УМ-3. Выписать программу которая выводит количество элементов в массиве X (массив состоит из 100 целых чисел и начинается с адреса 100), таких что  $X_i > i$  (нумерация элементов массива начинается с нуля).

9. Указать команды с ошибками и объяснить, в чём ошибка:

- а) MOV CS, AX
- б) LEA ES, [BP]
- в) ADC AH, [SI+BP]
- г) MUL [AL]
- д) JMP [SI+12]

10. Напечатать как знаковое число содержимое слова по абсолютному адресу 12345h.

### Семестр V

К зачету допускаются студенты, отработавшие все текущие задания семестра.

Зачёт выставляется по результатам итоговой оценки за все плановые текущие работы.

### Семестр VI

К зачету допускаются студенты, отработавшие все текущие задания семестра.

Выполнение задания “Форум” - 16 часов.

Оформление по правилам: Баула В.Г., Мещеряков Д.К. Учебное пособие по написанию программ при выполнении работ практикума на ЭВМ.

Зачет состоит в защите выполненного задания “Форум”.

#### **Задание «Форум»**

Используя технологии ASP.NET, ADO.NET реализовать веб-приложение "форум".

Приложение должно предоставлять основной функционал форума в сложившемся классическом его понимании в Интернете.

I. Реализовать подсистему учетных записей:

1. Предоставить возможность регистрации новых пользователей:

а. Логин пользователя должен быть уникальным в системе.

б. При регистрации проверять качество пароля (не должен быть слишком простым)

в. Также при регистрации исключить ситуацию автоматической регистрации пользователей с помощью зловредных программ (например, с помощью картинок с кодом, который необходимо ввести в поле для проверки).

г. При регистрации пользователь должен указать следующую информацию:

1. Желаемый логин\*.

2. Пароль\*.

3. Email\*.
4. Фамилию Имя
5. Возраст
6. Пол

\* указаны обязательные поля.

д. Зарегистрированный пользователь имеет право изменить всю свою информацию, кроме логина.

2. В системе должно существовать два вида пользователей: обычный пользователь и Администратор.

а. Обычный пользователь имеет право создавать темы в predetermined разделах.

б. Обычный пользователь имеет право просматривать все темы и оставлять в них сообщения.

в. Обычный пользователь может изменять параметры только им созданных тем, а также удалять только им созданные темы.

г. Администратор имеет право добавлять/изменять/удалять любое сообщение в теме, любую тему в разделе, любой раздел.

д. Администратор имеет право блокировать любого обычного пользователя в системе, после чего заблокированный пользователь должен быть уведомлен письмом и должен быть запрещен вход в систему для таких пользователей с выводом соответствующей ошибки.

## II. Реализовать подсистему непосредственно форума.

1. Главная страница форума представляет из себя список разделов (имя раздела, описание раздела).

2. При переходе в раздел должна отображаться страница со списком вложенных разделов и тем (имя темы, описание темы, кто создал, дата создания). Сначала список разделов в отдельной секции, потом список тем.

2.а Разделы могут быть вложенными. Т.е. внутри любого раздела может быть неограниченное кол-во вложенных разделов.

3. В каждой теме должен отображаться список сообщений (кто написал, что написал, когда написал).

4. При создании раздела (имеет право только Администратор) отображать форму создания с полями ввода имени и описания раздела. После сохранения перенаправлять на главную страницу.

5. При создании темы отображать форму с полями ввода для имени, описания темы. После сохранения перенаправлять на страницу соответствующего раздела со списками тем.

6. Для создания сообщения отображать форму с полем ввода для текста сообщения. После сохранения перенаправлять на страницу соответствующей темы.

## б) Система итогового контроля знаний

### Семестр VII

По результатам семинарских занятий проводится устный зачет с оценкой по билетам, составленным по программе дисциплины.

При выставлении итоговой оценки учитываются результаты текущего контроля.

## 5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### а) основная литература

#### Семестр I

1. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль. М., Наука, 1988.
2. Вирт Н. Алгоритмы и структуры данных. Спб., Невский проспект, 2005.
3. Пильщиков В.Н. Язык Паскаль: упражнения и задачи. М., Научный мир, 2004.
4. Шень А. Программирование: теоремы и задачи. М., МЦНМО, 2004.

#### Семестр II

1. Любимский Э.З., Мартынюк В.В., Трифонов Н.П. Программирование. М., Наука, 1980.
2. Пильщиков В.Н. Программирование на языке Ассемблера IBM PC. М., Диалог-МИФИ, 1994.

#### Семестр V

- 1 Шильдт Г. Полный справочник по C++, Пер. с англ.: -М : Издательский дом «Вильямс». 2004. - 800с.
- 2 Шильдт Г. C# Учебный курс. Пер. с англ.: Спб. ПитерЖ К.: Издательская группа ВHV. 2003. 512с.
- 3 Глушаков С.В., Лукошкина С. Н. Технология Java. – Харьков: Фолио, 2006. – 606с.
- 4 Пахомов Б. И. Visual C++ - 2008 для начинающих. – СПб.: БХВ Петербургу 2009. – 624с.
- 5 Фролов А.В. Фролов Г.В. Визуальное программирование приложений C#. М.: КУДИЦ-ОБРАЗ. – 2003. -512с.
- 6 Сошников Д. В. Программирование на F#. – М.: ДМК Пресс, 2011. – 192 с.: ил.

#### Семестр VI

1. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# /Пер. с англ. 2-е изд., испр. Спб.: Питер, 2008.
2. Просиз Дж. Програмирование для Microsoft .NET. /Пер. с англ. М.: Издательско-торговый дом «Русская Редакция». 2003.
3. Маклин С., Нафтел Дж., Уильям К. Microsoft .NET Remoting. /Пер. с англ. М.: Издательско-торговый дом «Русская Редакция». 2003.
4. Леве Дж. Создание служб WCF. Спб.: Питер. 2008.
5. Троелсен Э. Язык программирования C# 2005 и платформа .NET 2.0. М.: Вильямс. 2007.
6. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft.NET Framework 2.0 на языке C#. Microsoft Press. М.: Издательско-торговый дом «Русская редакция». 2007.
7. Нейгел К., Ивѐн Б. C# 2005 для профессионалов. М.: Диалектика. 2006.
8. Andrew Troelsen, Pro C# 2010 and the .NET 4 Platform, Fifth Edition

#### Семестр VII

1. Aho A., Sethi R., Ullman J. Compilers: principles, techniques and tools, N.Y., Addison-Wesley, 1986.

2. Грис Д. Построение компиляторов для цифровых вычислительных машин, М., "Мир", 1975.
3. Ахо А. Ульман Д. Теория синтаксического анализа, перевода и компиляции, М., "Мир", 1978.
4. Кнут Д. Искусство программирования для ЭВМ. Т.1. Основные алгоритмы, М., "Мир", 1976.

#### **б) дополнительная литература**

##### **Семестр I**

1. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. М., "Вильямс", 2000.
2. Кнут Д. Искусство программирования. Том 1. Основные алгоритмы. М., "Вильямс", 2005.
3. Кнут Д. Искусство программирования. Том 3. Сортировка и поиск. М., "Вильямс", 2005.
4. Корухова Л.С., Шура-Бура М.Р. Введение в алгоритмы. М., МГУ, 1997.
5. Пильщиков В.Н., Абрамов В.Г., Вылиток А.А., Горячая И.В. Машина Тьюринга и алгоритмы Маркова. Решение задач. М., МГУ, 2006.
6. Pascal ISO 7185:1990 – <http://www.moorecad.com/standardpascal/iso7185.pdf>

##### **Семестр II**

1. Скэлтон Л. Персональная ЭВМ IBM PC и XT. Программирование на языке Ассемблера. М., Радио и связь, 1991.
2. Абель П. Язык Ассемблера для IBM PC и программирования. М., Высшая школа, 1992.
3. Нортон П., Соухэ Д. Язык Ассемблера для IBM PC.

##### **Семестр V**

- 1 Вандервуд Д, и др. Шаблоны С++. Справочник разработчика. –М.: Издательский дом «Вильямс». – 2003. 554с.
- 2 Портянкин И. Swing эффективные пользовательские интерфейсы. Библиотека программиста.- Питер 2005. 360с.

##### **Семестр VI**

Баула В.Г., Мещеряков Д.К. Учебное пособие по написанию программ при выполнении работ практикума на ЭВМ.

##### **Семестр VII**

1. Г. Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ , 2-е издание. - М.:СПб, "Издательство Бином"- "Невский диалект",1998.
2. А. Элиенс. Принципы объектно-ориентированной разработки программ, 2-е издание. - М., Издательский дом "Вильямс", 2002.
3. Г. Шилдт. Самоучитель С++, 3-е издание. – СПб., БХВ-Петербург, 2001.
4. Б. Страуструп. Язык программирования С++, спец. издание. - М.:СПб, "Издательство Бином"- "Невский диалект",2001.
5. Т. Пратт, М. Зелковиц. Языки программирования. Разработка и реализация. – СПб, Издательский дом "Питер", 2002.

## в) Программное обеспечение и Интернет-ресурсы

### Интернет-ресурсы

- <http://cmcmsu.no-ip.info>

- электронная учебно-методическая система «Ownlibrari» кафедры программирования Филиала МГУ в г. Севастополе

### Семестр V

<http://www.nunit.org/>, <http://www.mbunit.com/>

<http://msdn.microsoft.com/en-us/library/default.aspx>

### Программное обеспечение

Операционные системы: Ubuntu Linux x 64; Debian 6 GNU/Linux; WindowsXP

1. Microsoft Office Std 2007 Rus OLP NL;
2. Windows Power Shell 1.0;
3. K-lite Codec Park 2.78 Standard;
4. Internet Explorer 8,0;
5. Protégé 3.4.7;
6. Mozilla Firefox;
7. Adobe Acrobat 9 PRO;
8. Adobe Reader X (10.0.1);
9. Adobe Reader 9.4.6;
10. Adobe Flash Player 11 Acrive X;
11. Core center;
12. Workstations Kaspersky WorkSpase Security Russian;
13. USB Disk Security v5.3.0\$
14. Nero 9,0;
15. Archiver WinRAR 3.90;
16. Borland Pascal;
17. FreePascal 2.2.4;
18. Eclipse 3.3.2;
19. FAR Manager;
20. Microsoft Visual C++ 2008 Express Edition;
21. Microsoft Visual Studio 2010 Ultimate;
22. Microsoft SQL Server 2008;
23. Download Master;
24. WinDjView;
25. ICQ 6.5;
26. Skype;
27. Java(TM) Se

## 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

- библиотека Филиала МГУ в г. Севастополе;
- библиотека кафедры программирования Филиала МГУ в г. Севастополе;
- лекционные аудитории;
- специализированные компьютерные классы с доступом к Интернет-ресурсам с любого компьютера:

### Серверы

Acer altos 500 (P3-500Mhz/Ram 128M (Internet-сервер) – 2 шт.;  
Acer altos 11000 (P3-700Mhz/Ram 256M (Windows-сервер) – 2 шт.;  
2x Quad Core Xeon E5410, MB Intel 8Gb,GF GTX260, 2x 1000Gb, Case + ИБП APC  
Smart-UPS SC 1100V (Linux - сервер)

### Коммутационный шкаф

SuperStack II HUB 24 ports;  
SuperStack II HUB 24 ports

### Компьютеры

Intel Core Quad Q6600,iP45, 4Gb,GF GTX260,500Gb+1Tb,DVD, Case, Keyb, Mouse,  
WinXPpro – 1 шт.;  
Intel C2D E7400,iG31,2Gb, intVideo,500Gb,DVD,Case, Keyb, Mouse, WinXPpro – 15  
шт.;  
PENTIUM-IV 2.4 GHz 512 Mб RAM – 5 шт.;  
PENTIUM-IV 3.06 GHz 512 Mб RAM – 1 шт.

Вебкамера – 2 шт.

Wi-fi – точка;

Акустическая система F&D 611;

Гарнитура Genius HS-04A – 16 шт.;

Мультимедийный проектор Aser;

Микрофон Genius MIC-01A для Skype;

Принтер HP LazerJet P1005;

Сканер HP scanjet 3500c.

**ПОЛОЖЕНИЕ**  
**о проведении пересдач задолженностей студентов по итоговым аттестациям**  
**на кафедре программирования Филиала МГУ в г. Севастополе**  
**прикомандированными преподавателями с факультета ВМК МГУ**  
**протокол №5 заседания кафедры от 10 апреля 2012 г.**

Настоящее положение регулирует порядок проведения пересдач задолженностей студентами факультета Компьютерной математики на кафедре программирования Филиала МГУ в г. Севастополе прикомандированными преподавателями с факультета ВМК МГУ.

1. Пересдача задолженностей принимается по графику пересдач задолженностей, установленному кафедрой программирования в принятые ректоратом МГУ имени М.В.Ломоносова сроки, в отдельных случаях - в сроки установленные комиссией по студенческим делам.
2. График пересдач задолженностей сообщается в учебный отдел в виде служебной записки и преподавателю факультета ВМК МГУ, по дисциплине которого проводится пересдача.
3. Пересдачи экзаменов (зачётов) в отсутствие преподавателей факультета ВМК в Филиале МГУ в г. Севастополе проводятся в письменной форме, независимо от того в какой форме проводился основной экзамен (зачёт).
4. Вариант письменной экзаменационной (зачётной) работы для пересдачи составляет преподаватель, проводивший основной экзамен. Также преподаватель указывает требования к проведению экзамена (сколько времени даётся на написание работы, какими материалами разрешается пользоваться студенту при написании работы и др.)
5. Кафедрой программирования назначаются местные преподаватели кафедры, которые проводят пересдачу в установленные даты и время в соответствии с графиком пересдач по присланному варианту с выполнением всех требований к проведению.
6. Написанные студентами при пересдаче экзаменационные (зачётные) работы сканируются по окончании пересдачи и пересылаются по электронной почте на проверку преподавателю факультета ВМК МГУ.
7. Преподавателю ВМК МГУ пересылается скан ведомости пересдачи для выставления отметок по результатам пересдачи.
8. Преподаватель ВМК присылает в Филиал скан заполненной им ведомости пересдачи и сканы проверенных им работ.
9. Полученный скан ведомости подписывается зам. зав. кафедрой.
10. Сканы ведомости пересдачи сдаются в учебный отдел Филиала.
11. Кафедра знакомит студентов, писавших работу на пересдаче, с результатами пересдачи.
12. При возникновении у студентов, писавших работу на пересдаче, вопросов по результатам проверки, студент может обратиться к преподавателю ВМК лично по электронной почте (скайпу). Свои координаты для консультаций с ним преподаватель сообщает студентам на первой лекции курса.