

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное образовательное учреждение  
высшего профессионального образования  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет вычислительной математики и кибернетики

УТВЕРЖДАЮ

Декан факультета ВМК

\_\_\_\_\_ Е.И. Моисеев

«\_\_\_\_\_» \_\_\_\_\_ 2015

**Учебно-методический комплекс**  
**«Практикум на ЭВМ», 4 семестр**

Направление подготовки

010300 «*Фундаментальная информатика и информационные технологии*»

Бакалавр

Профиль подготовки (общий)

\_\_\_\_\_

Квалификация (степень) выпускника

Бакалавр

Форма обучения

очная

Москва

2015

## Содержание

1. Цели и задачи освоения дисциплины
2. Место дисциплины в структуре ООП ВПО
3. Требования к результатам освоения содержания дисциплины
4. Структура дисциплины (модуля) и ее место в учебном плане
  - 4.1 Тематический план курса
  - 4.2 Структура дисциплины по видам работ
  - 4.3 Лабораторные работы
  - 4.4 Курсовой проект (курсовая работа, расчетно-графическое задание, реферат, контрольная работа)
  - 4.5 Консультации
  - 4.6 Интерактивные образовательные технологии, используемые в аудиторных занятиях
5. Содержание дисциплины «Практикум на ЭВМ» (4 семестр). План семинарских занятий.
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации
  - 6.1. Контрольная работа № 1. «Язык Си++» (образец)
  - 6.2 Контрольная работа №2. «Формальные грамматики и языки. Элементы теории трансляции» (образец)
7. Оценочные средства рубежного контроля
8. Учебно-методическое и информационное обеспечение дисциплины
9. Материально-техническое обеспечение дисциплины

## Рабочая программа дисциплины **«Практикум на ЭВМ»**

/ составитель к.ф.-м.н. Полякова И.Н.

Рабочая программа предназначена для преподавания дисциплины **«Практикум на ЭВМ»** базовой части ЕН цикла студентам очной формы обучения по направлению подготовки **«010300 Фундаментальные информатика и информационные технологии»** в 4 семестре.

Рабочая программа составлена с учетом Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки, утвержденного приказом Министерства образования и науки Российской Федерации от "8" декабря 2009 г. № 712, а также образовательного стандарта МГУ бакалавр по направлению **«010300 Фундаментальные информатика и информационные технологии»**.

Составитель:

к.ф.-м.н. Полякова Ирина Николаевна

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

Данная дисциплина читается в поддержку основного лекционного курса «Системы программирования» базовой части ЕН цикла . Целью освоения дисциплины является получение навыков практического программирования на языке C++, а также умение решать задачи по теории формальных языков и грамматик, на которых базируются современные трансляторы языков программирования. Отдельной задачей ставится умение написать анализатор для модельного языка программирования с применением объектно-ориентированного языка программирования C++ и элементов теории трансляции.

## ***2. Место дисциплины в структуре ООП бакалавриата***

Дисциплина относится к базовой части профессионального цикла. Содержание курса определяется образовательным стандартом МГУ высшего профессионального образования по направлению 010300 Фундаментальные информатика и информационные технологии.

Дисциплина "Практикум на ЭВМ" состоит из двух частей:

1. Семинарские занятия в поддержку курса **«Системы программирования»**.

2. Выполнение практических заданий по написанию и отладке программ на компьютерах.

Логически и содержательно-методически дисциплина поддерживает базовый лекционный курс. **«Системы программирования»**. Изучение курса "Практикум на ЭВМ" опирается на знания, полученные студентами после прослушивания курсов «Основы программирования», «Архитектура ЭВМ и язык ассемблера» и «Объектно-ориентированное программирование».

Освоение данной дисциплины (модуля) необходимо студентам (независимо от их дальнейшей специализации) для закрепления навыков программирования в объектно-ориентированном стиле, весьма актуальном на сегодняшний день, а также получения навыков в построении анализаторов языков программирования.

### **3. Требования к результатам освоения содержания дисциплины**

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВПО по данному направлению:

**общекультурных (ОК):** владеть основными методами, способами и средствами получения, хранения, переработки информации, иметь навыки работы с компьютером как средством управления информацией ОК-(12);

**общенаучных (ОНК):** владеть фундаментальными разделами математики и информатики, необходимыми для решения научно-исследовательских и практических задач в профессиональной области (ОНК-б);

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

**обще-профессиональных (ПК):** способность решать задачи производственной и технологической деятельности, включая разработку алгоритмических и программных решений в области системного и прикладного программирования (ПК-9);

**Способность понимать и применять в исследовательской и прикладной деятельности методы объектно-ориентированного программирования и методы теории трансляции**

**в инновационной деятельности:**

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

В результате освоения дисциплины студент должен:

**знать** основы теории формальных языков и грамматик и теории трансляции, понятия объектно-ориентированного программирования.

**уметь применять на практике** навыки практического программирования на языке С++, а также решать задачи по теории формальных языков и грамматик, на которых базируются современные трансляторы языков программирования;

**понимать и применять на практике** компьютерные технологии для решения различных задач в объектно-ориентированном стиле;

**уметь**

- находить, анализировать и контекстно обрабатывать научно-техническую информацию;

- извлекать полезную научно-техническую информацию из электронных библиотек, реферативных журналов;

- демонстрировать способность к анализу и синтезу;

- демонстрировать способность к письменной и устной коммуникации на русском языке;

- публично представить собственные и известные научные результаты;

**владеть**

- методами теории трансляции, объектно-ориентированного программирования;

- навыками решения практических задач теории трансляции;

- навыками решения практических задач, связанных с объектно-ориентированным программированием;

#### **4. Структура дисциплины (модуля) и ее место в учебном плане**

##### **4.1 Тематический план курса**

№	Название темы	Аудиторные занятия - семинары (часы)	Самостоятельная работа студента
1.	<b>Языки и системы программирования, поддерживающие ООП. Язык программирования Си++.</b>	18	18
2.	<b>Элементы формальных грамматик и теории трансляции.</b>	14	22
	Всего	32	40

Общая трудоемкость дисциплины составляет 2 зачетные единицы (72 часа).

Семинары – 32 часа, самостоятельная работа - 40 часов.

#### 4.2. Структура дисциплины по видам работ

№ п/п	Раздел Дисциплины	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)		Формы текущего контроля успеваемости ( <i>по неделям семестра</i> )  Форма промежуточной аттестации ( <i>по семестрам</i> )
			Семинары	Сам	
1.1	Обзор основных свойств и возможностей объектно-ориентированного программирования на языке Си++, рассмотренных в курсе “ООП” в 3 семестре. Работа с динамической памятью.	1-2	4	4	Самостоятельная работа, индивидуальный опрос
1.2	Перегрузка функций. Алгоритм выбора перегруженной функции.	3	2	2	Самостоятельная работа, индивидуальный опрос
1.3	Наследование единичное, множественное	4-5	4	4	Самостоятельная работа, индивидуальный опрос
1.4	Исключения в С++	6-7	4	4	Самостоятельная работа, индивидуальный

					опрос
1.5	Динамическая идентификация типа. Параметрический полиморфизм. Шаблоны функций и классов .	8	2	2	Самостоятельная работа, индивидуальный опрос
1.6	Решение типовых задач по курсу.	9	2	2	Контрольная работа по темам пункта 1.
2.1	Выдача задания практикума	10	2	4	
2.2	Основные понятия теории формальных грамматик и языков	11	2	3	Самостоятельная работа, индивидуальный опрос
2.3	Регулярные и автоматные грамматики.	12	2	3	Самостоятельная работа, индивидуальный опрос
2.4	Метод рекурсивного спуска.	13	2	3	Самостоятельная работа, индивидуальный опрос
2.5	Грамматики с действиями. Синтаксически управляемый перевод.	14	2	3	Самостоятельная работа, индивидуальный опрос
2.6	ПОЛИЗ, алгоритм интерпретации ПОЛИЗ.	15	2	3	Самостоятельная работа, индивидуальный опрос

2.7	Решение типовых задач по курсу.	16	2	3	Контрольная работа по темам пункта 2, прием задания Практикума
-----	---------------------------------	----	---	---	--

#### ***4.3. Лабораторные работы***

Лабораторные работы не предусмотрены учебным планом

#### ***4.4. Курсовой проект (курсовая работа, расчетно-графическое задание, реферат, контрольная работа)***

Курсовая работа не предусмотрена учебным планом

#### ***4.5. Консультации***

Преподаватели, ведущие практические занятия, периодически в течение семестра проводят консультации по дисциплине. Также на консультациях, по мере необходимости, более подробно разбирается выполнение задания Практикума на компьютерах по написанию анализатора.

#### ***4.6. Интерактивные образовательные технологии, используемые в аудиторных занятиях***

Используются традиционные технологии проведения семинарских занятий в аудиториях и на компьютерах. Все методические материалы для прохождения дисциплины отражены на сайте в Интернете.

#### ***5. Содержание дисциплины «Практикум на ЭВМ» (4 семестр). План семинарских занятий.***

Практикум на ЭВМ включает в себя семинары и практические занятия на компьютерах. Практические занятия посвящены совершенствованию навыков студентов в программировании на языке Си++. На практических занятиях студенты должны предъявить преподавателю написанные самостоятельно программы решения задач и, в конце семестра, самостоятельно выполненное задание практикума - лексический и синтаксический анализаторы (фрагмент из задания [5] основной литературы “Интерпретатор модельного языка программирования” Руденко Т.В. <http://cmcmsu.no-p.info/download/model.lang.practical.task.pdf> или аналогичное по сложности задание).

*Номера задач* с префиксом «С» даются по задачнику [3] основной литературы (И.А. Волкова, А.А. Вылиток, Л.Е. Карпов. Сборник задач и упражнений по языку Си++); с префиксом «К» - по задачнику [4] основной литературы (Ю.С. Корухова. Сборник задач и упражнений по языку Си++); с префиксом «В» – по методическому пособию [6] основной литературы (И.А. Волкова, А.А. Вылиток, Т.В. Руденко. Формальные грамматики и языки. Элементы теории трансляции).

## **Семинар № 1-2. «Обзор основных свойств и возможностей объектно-ориентированного программирования на языке Си++. Работа с динамической памятью»**

### Семинар №1:

1. О занятиях по практикуму в семестре (темы аудиторных занятий, задание на машине и т.п.)
2. Задачи: С-1.1, С-1.2, С-1.3(а), С-1.5, С-1.11, разбор примеров из лекций
3. Домашнее задание: С-1.3(б) , С-1.7, С-1.10, С-1.13

### Семинар № 2.

Практическое занятие в компьютерном классе. Решение задач типа К-1.9

Домашнее задание: С-1.16 , С-1.20, С-1.22, С-6.1, С-6.3(а), 6-4(а)

## **Семинар № 3. «Перегрузка функций»**

- ✓ Алгоритм поиска оптимально отождествляемой функции (для одного параметра):
  - Точное отождествление
  - Отождествление при помощи расширений
  - Отождествление при помощи других стандартных преобразований
  - Отождествление при помощи преобразований, определенных пользователем
  - Отождествление по ...
- ✓ Разрешение вызова для функций с несколькими параметрами

Задачи: С-2.1, С-2.2(а,б), С-2.3(а), С-6.7(а), С-6.8(а), разбор примеров из лекций

Домашнее задание: С-2.2(с,д), С-2.3(б,с), С-6.7(б), С-6.8(б)

## **Семинар № 4-5. « Наследование единичное, множественное»**

### Семинар № 4

- Функции-члены – унаследованные и замещенные.
- Конструкторы и деструкторы: порядок вызова, передача параметров
- Виртуальные функции:

- Указатели на базовый и производный классы, преобразование указателей
- Динамическое связывание
- Неоднозначность из-за неоднократно повторяющегося базового класса, разрешение неоднозначности с помощью операции ::
- Виртуальные базовые классы
- Неоднозначность из-за совпадающих имен в различных базовых классах, разрешение неоднозначности с помощью операции ::
- Однозначность, совместное использование, доступ

Задачи: С-3.1(а), С-3.2(а,е), С-3.3, С-4.2, 4.12, разбор примеров из лекций

Домашнее задание: С-3.1(б,с), С-3.2(б), С-3.4

### Семинар № 5

Практическое занятие в компьютерном классе. Решение задач типа К-3.3

Домашнее задание: С-4.4, С-4.6, С-4.9, С-4.14

### **Семинар № 6-7. «Исключения в Си++»**

#### Семинар № 6

- Общая схема обработки исключений: *try – throw – catch*
- Функции *terminate()* и *unexpected()*
- Правило выбора обработчика исключения, свертка стека
- Повторное возбуждение исключения

Задачи: С-5.1, С-5.4, С-5.7, К- 4.5(в), разбор примеров из лекций

Домашнее задание: С-5.2, С-5.5, С-5.8, К- 4.1

#### Семинар № 7

Практическое занятие в компьютерном классе. Решение задач типа К-4.6,

К -4.8(а,в)

Домашнее задание: С-5.10, С-5.11, С-5.13, С-5.14

## **Семинар № 8. «Динамическая идентификация типа. Параметрический полиморфизм. Шаблоны функций и классов»**

- Динамическая идентификация и приведение типов
- Средства RTTI
- Шаблоны функций
- Шаблоны классов

Задачи: К-5.1, С-7.1, С-7.2, С-8.3, С-8.6 (пункты по выбору)

Домашнее задание: К-5.2, К-5.4, С-7.3, С-7.6, С-8.2, С-8.6

## **Семинар № 9. «Контрольная работа № 1»**

Образец контрольной работы по Си++ - см. раздел 6.1

## **Семинар № 10. «Выдача задания практикума»**

Необходимо написать лексический и синтаксический анализаторы (фрагмент из задания [5] основной литературы “Интерпретатор модельного языка программирования” Руденко Т.В. <http://cmcmsu.nor.info/download/model.lang.practical.task.pdf>) или аналогичное по сложности задание).

## **Семинар № 11. «Основные понятия теории формальных грамматик и языков»**

- Основные определения
- Классификация по Хомскому
- Соотношения между типами грамматик и языков
- Определение языка по заданной грамматике и построение грамматики по заданному языку
- Проблемы однозначности и эквивалентности грамматик
- Задача разбора, дерево вывода
- Алгоритм приведения грамматик

Задачи: В-1.1(b), В-1.3(a,b,c,d,e,f), В-1.4(f,g), В-1.8(b), В-1.11(a), В-1.20, В-1.21, В-1.26(a), В-1.27

Домашнее задание: В-1.2, В-1.3(i, j,n,p), В-1.4(a,b,c), В-1.5, В-1.6(b), В-1.9, В-1.11(b), В-1.12, В-1.23, В-1.26(b)

## **Семинар № 12. «Регулярные и автоматные грамматики»**

- Определение регулярной грамматики
- Автоматные грамматики
- Конечные автоматы и диаграммы состояний (ДС)
- Анализаторы по ДС
- Недетерминированный разбор, алгоритм преобразования НКА в ДКА

Задачи: В-2.1, В-2.2, В-2.4(a), В-2.10(a), В-2.12(a,b,n,o)

Домашнее задание: В-2.3, В-2.4(b), В-2.10(b,c), В-2.11, В-2.12(c,d,k,r)

## **Семинар № 13. «Метод рекурсивного спуска»**

- Подкласс грамматик, к которому применим рекурсивный спуск
- Критерий применимости метода рекурсивного спуска.
- Преобразования грамматик, эквивалентность рассматриваемых преобразований.

Задачи: В-3.1(a,b,c,d,e), В-3.2(a,c), В-3.3(a,b,j)

Домашнее задание: В-3.1(f,g,h,i,j), В-3.2(b,d), В-3.3(c,d,h)

## **Семинар № 14. «Граматики с действиями. Синтаксически управляемый перевод»**

Задачи: В-3.4, В-3.5, В-3.7, В-3.8, В-4.1(a), В-4.2(a)

Домашнее задание: В-3.6, В-3.9, В-3.10, В-4.1(b), В-4.2(d)

## **Семинар № 15. «ПОЛИЗ – язык внутреннего представления программ»**

- ПОЛИЗ выражений
- ПОЛИЗ операторов модельного языка
- Интерпретация ПОЛИЗ

Задачи: В-5.1(a,b), В-5.2(a,b), В-5.3(a), В-5.4(a,b,c), В-4.3(a), В-5.6, В-5.13

Домашнее задание: В-5.1(c,d,g), В-5.2(c,d,g), В-5.3(b), В-5.4(e,f,i,j), В-4.3(c), В-5.7, В-5.14

## **Семинар № 16. «Контрольная работа № 2 и сдача задания практикума»**

Образец контрольной работы (на 45 минут) по теории формальных грамматик - см. раздел 6.2

Вторая половина занятия – сдача задания практикума. Сдача задания подразумевает умение связно отвечать на вопросы по существу программы, а также способность вносить в свою программу дополнения и изменения по требованию преподавателя.

## 6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации

### 6.1. Контрольная работа № 1. «Язык Си++» (образец)

1. Что напечатает следующая программа?

```
class I { int i;
    public: I() : i(6) { cout << "owl" << endl; }
           I(int a) : i(a) { cout << "sheep " << i << endl; }
           I(const I & other) : i(other.i) { cout << "horse " << i << endl; }
           ~I() { cout << "wolf" << endl; }
           int Get() { return i; }
           void operator*=(const I & op) { i*=op.i; } };

void f(I x, I & y) {
    x *= 1; y *= x; }

int main() { I i1; I i2(3);
            i1 *= 7;
            f(i1, i2);
            cout << i1.Get() << ' ' << i2.Get() << endl;
            return 0; }
```

2. Объект  $x$  принадлежит классу  $X$ , для которого определены конструктор преобразования типа `int` в класс  $X$  и функция преобразования класса  $X$  в тип `int`. .Описать прототипы **двух** перегруженных функций **h** из некоторой области видимости, для которых будут верны следующие обращения к ним:

```
h ("333"); h (x, 0); h (1, "dd ); h (0,0); h(x, "nn");
```

3. Найти ошибки в программе и объяснить, в чем они заключаются.

```
class table{ int size; int priority;
    public: table(int s=0,int p):size(s),priority(p){ }
    virtual void print()=0; };

class stud_table: public table{ char * name; int gr;
    public: void print(){cout<<"students table"<<endl;}
    ~stud_table(){delete []name;} };

class asp_table:protected table{ char* thesis; };

int main(){ table t; stud_table st; table * tp = &st;
            tp = new asp_table();
            stud_table * stp = &st;
            cout<<"Program"<<endl; return 0; }
```

4. Что будет выдано в стандартный канал вывода при работе следующей программы?

```
struct X;
void f(X & x, int n);
int const P = 1; int const Q = 1; int const R = 1;
struct X { X() { try { f(*this, -1);
                    cout << 1 << endl; }
          catch (X) { cout << 2 << endl; }
          catch (int) { cout << 3 << endl; } }
  X(X &) { cout << 4 << endl; }
  ~X () { cout << 5 << endl; }
};
struct Y: X { Y () { f(*this, 1);
                  cout << 6 << endl; }
  Y (Y &) { cout << 7 << endl; }
  ~Y () { cout << 8 << endl; }
};
void f (X & x, int n) { try { if (n < 0) throw x;
                             if (n > 0) throw 1;
                             cout << 9 << endl; }
  catch (int) { cout << 10 << endl; }
  catch (X& a) { cout << 11 << endl;
                f(a, 0);
                cout << 12 << endl;
                throw; }
};
int main() { try { Y a; }
  catch (...) { cout << 13 << endl;
               return 0; }
  cout << 14 << endl;
  return 0;
}
```

5. Для приведённой ниже программы описать функцию  $f()$ , которая, получая в качестве параметра ссылку на объект базового класса  $A$ , возвращает ссылку на объект производного класса  $C$ , полученную наиболее безопасным образом, а в случае невозможности приведения типов корректно завершает программу.

```
struct A { virtual void z () {} };
struct B: A { };
struct C: B { int x;
  C (int n = 3) { x = n; } };

```

```

C & f (A & ra);
int main () { C c, & pc = f (& c);
              cout << pc.x << endl;
              return 0;
            }

```

6. Для класса рациональных дробей с числителями и знаменателями некоторого интегрального типа

```

Template <class T> class fr { T n; T d; ... };

```

описать два варианта (методом класса и функцией-другом этого класса) реализации вне класса операций '\*', выполняющей умножение одной рациональной дроби на другую.

## 6.2 Контрольная работа №2. «Формальные грамматики и языки. Элементы теории трансляции» (образец)

1. а) Построить приведенную грамматику, эквивалентную заданной грамматике G:

$$S \rightarrow aAb \mid CB$$

$$B \rightarrow bB$$

$$A \rightarrow aA \mid D$$

$$C \rightarrow cCc \mid B \mid c$$

$$D \rightarrow d$$

б) Каков тип получившейся грамматики?

в) Какой язык порождает грамматика?

г) Каков тип этого языка?

*Для типов грамматики и языка указать максимально возможный номер по Хомскому.*

2. Дана автоматная левосторонняя грамматика  $G$ :

$$S \rightarrow C \perp$$

$$A \rightarrow Aa \mid a$$

$$B \rightarrow Aa \mid b$$

$$C \rightarrow Ba$$

- а) Детерминирован ли разбор по этой грамматике? **Обосновать ответ.**
- б) Построить по грамматике конечный автомат и преобразовать его к детерминированному виду по алгоритму НКА  $\rightarrow$  ДКА
- в) Построить по получившемуся ДКА праволинейную автоматную грамматику.
- г) Эквивалентны ли исходная и получившаяся автоматные грамматики? **Обосновать ответ.**

3. Дана КС-грамматика  $G$ :

$$S \rightarrow A \mid B$$

$$A \rightarrow aA \mid \varepsilon$$

$$B \rightarrow bB \mid cC \mid \varepsilon$$

$$C \rightarrow aABC \mid c$$

- а) Преобразовать грамматику  $G$  к неукорачивающей КС-грамматике с помощью алгоритма устранения пустых правых частей в КС грамматике (КС  $\rightarrow$  НКС).
- б) Применим ли метод рекурсивного спуска к исходной грамматике?  
**Ответ обосновать.**

4. Построить грамматику выражений, содержащих

- целочисленные бинарные операции  $+$  и  $-$
- односимвольные целые числа **1** и **3**

Добавить в грамматику действия (только вида `cout <<"символ";`) по переводу заданных выражений в ПОЛИЗ во время анализа РС-методом.

Приоритет операций + и – должен быть задан построенной грамматикой таким образом, чтобы ПОЛИЗ выражения:

$$3 - 1 + 1$$

совпадал с ПОЛИЗом выражения со скобками:

$$(3 - (1 + 1))$$

## 7. *Оценочные средства рубежного контроля*

*Необходимые (для получения зачета) практические навыки*

- Определение типа по Хомскому заданной грамматики.
- Определение типа языка, порождаемого заданной грамматикой.
- Построение грамматики [определенного типа], порождающей заданный язык.
- Определение языка, порождаемого заданной грамматикой.
- Построение приведенной грамматики.
- Устранение пустых правых частей заданной КС-грамматики.
- Построение ДС конечного автомата по заданной левوليнейной грамматике.
- Построение анализатора на С++ по заданной ДС.
- Восстановление левوليнейной регулярной грамматики, порождающей язык, распознаваемый конечным автоматом, заданным в виде ДС.
- Восстановление левوليнейной регулярной грамматики по заданному тексту анализатора на С++.
- Построение ДС по праволинейной грамматике и построение праволинейной грамматики по заданной ДС.
- Преобразование праволинейной грамматики в эквивалентную левوليнейную с помощью ДС.
- Преобразование НКА в эквивалентный ДКА с помощью соответствующего алгоритма.
- Определение и обоснование применимости метода рекурсивного спуска к заданной КС-грамматике.
- Построение (на С++) анализатора методом рекурсивного спуска для заданной КС-грамматики.
- Восстановление КС-грамматики по заданному анализатору, построенному методом рекурсивного спуска.
- Определение языка, порождаемого грамматикой с действиями.
- Дополнение заданной КС-грамматики действиями, позволяющими учесть дополнительные ограничения на цепочки определяемого ею языка.
- Вставка в заданную (или построенную) грамматику языка  $L_1$  действий по переводу цепочек этого языка в цепочки языка  $L_1$  по заданному закону соответствия между цепочками (т. е. реализовать формальный перевод).
- Определение языков  $L_1$  и  $L_1$  по заданной грамматике с действиями, реализующей формальный перевод языка  $L_1$  в язык  $L_1$ .
- Запись в ПОЛИЗе заданного фрагмента программы на заданном языке.

- Восстановление текста на заданном языке заданного фрагмента программы в ПОЛИЗе.
- Определение, является ли данная запись ПОЛИЗом заданной конструкции.
- Вставка в грамматику, порождающую некоторую конструкцию языка программирования, действий, осуществляющих синтаксически управляемый перевод этой конструкции в ПОЛИЗ при анализе РС-методом.

**‘+’ умение писать и анализировать программы на языке Си++ в рамках пройденной тематики**

**Зачёт с оценкой.**

*Зачёт проставляется по результатам освоения студентом дисциплины (контрольная работа №1 и контрольная работа № 2) и сдачи задания практикума.*

*Для не отчитавшихся в течение семестра предусмотрена зачетная работа по типу контрольной №1, контрольной № 2, а также решение задач на компьютерах – написание мини-анализаторов по регулярным и/или контекстно-свободным грамматикам (методом рекурсивного спуска).*

## 8. Учебно-методическое и информационное обеспечение дисциплины)

а) основная литература:

1. И. А. Волкова, А. В. Иванов, Л. Е. Карпов. Основы объектно-ориентированного программирования. Язык программирования С++. Учебное пособие для студентов 2 курса. — М.: Издательский отдел факультета ВМК МГУ, 2011.

*Электронная версия:* <http://cmcmsu.no-ip.info/download/cpp.base.oop.pdf>

2. А.В.Столяров. Введение в язык С++. Учебное пособие.-3 изд. – М.МАКС Пресс, 2012.

*Электронная версия:*<http://www.stolyarov.info/books/pdf/cppintro3.pdf>

3. И. А. Волкова, А. А. Вылиток, Л.Е. Карпов. Сборник задач и упражнений по языку С++. Учебное пособие для студентов II курса.— М.: Издательский отдел факультета ВМК МГУ, 2013

*Электронная версия:* <http://cmcmsu.no-ip.info/download/cpp.tasks.2013.pdf>

4. Ю.С. Корухова. Сборник задач и упражнений по языку С++. Учебное пособие для студентов-бакалавров II курса, обучающихся по направлению «Информационные технологии». — М.: Издательский отдел факультета ВМК МГУ, 2009

*Электронная версия:*

<http://cmcmsu.no-ip.info/download/korukhova.cpp.tasks.pdf>

5. Руденко Т.В. Интерпретатор модельного языка программирования. (<http://cmcmsu.no-ip.info/download/model.lang.practical.task.pdf>)

6. И. А. Волкова, А. А. Вылиток, Т. В. Руденко. Формальные грамматики и языки. Элементы теории трансляции (3-е издание). — М.: Изд-во МГУ, 2009.

*Электронная версия:*

<http://cmcmsu.no-ip.info/download/formal.grammars.and.languages.2009.pdf>

б) дополнительная литература:

1. Б. Страуструп. Язык программирования С++. Специальное издание. — М.: Издательство «БИНОМ», 2011. (Шифр в библиотеке МГУ: 5ВГ66 С-835)
2. Г. Шилдт. Самоучитель С++. 3-е изд. — СПб: БХВ-Петербург, 2002. (Шифр в библиотеке МГУ: 5ВГ66 Ш-576)
3. Страуструп Б. Программирование: принципы и практика использования С++.: Пер. с англ. – М. ООО «И.Д.Вильямс», 2011.
4. Страуструп Б. Дизайн и эволюция С++. Пер. с англ. – М.: ДМК Пресс, 2006.
5. Д. Грис. Конструирование компиляторов для цифровых вычислительных машин. — М.: Мир, 1975.
6. А. Ахо, Дж. Ульман. Теория синтаксического анализа, перевода и компиляции, т.1,2 — М.: Мир, 1979.
7. Мейерс С. Эффективное использование С++. 50 рекомендаций по улучшению ваших программ и проектов. Питер, ДМК Пресс, Москва, 2006
8. Мейерс С. Эффективное использование С++. 35 новых рекомендаций по улучшению ваших программ и проектов. Питер, ДМК Пресс, Москва, 2012

## в) Программное обеспечение и Интернет-ресурсы.

Для проведения практических занятий необходимо наличие компилятора языка C++.

Кроме того, студентам предлагается искать дополнительную информацию на сайтах, посвященных объектно-ориентированному программированию на языке C++, в частности на сайте комитета по стандартизации языка C++ (The C++ Standards Committee) <http://www.open-std.org/JTC1/SC22/WG21/>, на сайтах, посвященных системам программирования. Также предлагается использовать информацию, размещённую на сайте <http://al.cs.msu.su>

## ***9. Материально-техническое обеспечение дисциплины***

Наличие литературы в библиотеке и компьютер для проведения практических занятий.

Программа составлена в соответствии с требованиями ФГОС ВПО, ОС МГУ «Фундаментальные информатика и информационные технологии», с учетом рекомендаций Примерной основной образовательной программы (ПрООП) по направлению 010300 «Фундаментальные информатика и информационные технологии», бакалавриат.