

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

«_____» _____ 2014

Учебно-методический комплекс
«Системы программирования»

Направление подготовки
010300 «*Фундаментальные информатика и информационные технологии*»
Бакалавр

Квалификация (степень) выпускника
Бакалавр

Форма обучения
очная

Москва

2014

Рабочая программа дисциплины **«Системы программирования»**

/ составитель к.ф.м.н. Полякова И.Н.

Рабочая программа предназначена для преподавания дисциплины **«Системы программирования»** 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

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 «Системы программирования».
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации
 - 6.1. Коллоквиум № 1
 - 6.2 Коллоквиум № .
7. Оценочные средства рубежного контроля
 - 7.1. Вопросы к экзамену
8. Учебно-методическое и информационное обеспечение дисциплины
9. Материально-техническое обеспечение дисциплины

1. Цели освоения дисциплины

Целью освоения дисциплины является знакомство с современными системами программирования, значительная часть которых базируется на объектно-ориентированных языках программирования.

В частности, ставятся следующие задачи:

1) продолжить изучение возможностей объектно-ориентированного языка программирования C++, начатого в 3 семестре в курсе «Объектно-ориентированное программирование»;

2) изучить основы теории формальных языков и грамматик;

3) познакомиться с современными системами программирования (состав, назначение, схема работы);

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.

Продолжается начатое в 3 семестре изучение объектно-ориентированного языка программирования C++, изучаются современные системы программирования, а также теория формальных грамматик и языков, необходимая при построении трансляторов – основного компонента систем программирования. Изучение 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 актуальном на сегодняшний день объектно-ориентированном стиле, а также получения навыков работы с современными системами программирования.

3. Требования к результатам освоения содержания дисциплины

В результате освоения дисциплины выпускник должен обладать следующей **обще-профессиональной компетенцией (ПК)**.

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

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

Способность понимать и применять в исследовательской и прикладной деятельности методы объектно-ориентированного программирования и методы теории трансляции, в том числе:

- способность понимать концепции и использовать на практике следующие базовые знания:
 - **на уровне технической грамотности:** постулаты объектно-ориентированного программирования, теория формальных языков и грамматик;
 - **на уровне понимания концепций, способности их использования:**
 - полиморфизм (динамический – механизм виртуальных функций, статический – перегрузка функций и операций, параметрический – механизм шаблонов)
 - наследование (единичное и множественное);
 - аппарат исключений,
 - аппарат RTTI;
 - метод преобразования недетерминированного конечного автомата к детерминированному;
 - метод рекурсивного спуска;
- способность понимать и самостоятельно разрабатывать алгоритмы для решения задач системного и прикладного программирования, исследования математических, информационных и имитационных моделей по тематике выполняемых работ;
- понимание концепций и способность использовать в профессиональной деятельности основные законы естественнонаучных дисциплин;

в) в результате освоения дисциплины студент должен:

знать основы теории формальных языков и грамматик и теории трансляции, основные концепции объектно-ориентированного программирования.

уметь применять на практике основные методы объектно-ориентированной парадигмы и теории трансляции;

понимать и применять на практике основные методы теории трансляции, 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;

- очно представлять математические знания в устной форме;

владеть

- навыками решения практических задач теории трансляции, задач объектно-ориентированного программирования;

- методами теории трансляции, объектно-ориентированного программирования;

- проблемно-задачной формой представления естественнонаучных знаний.

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

4.1 Тематический план курса

№	Название темы	Аудиторные занятия - лекции (часы)	Самостоятельная работа студента
1.	Языки и системы программирования, поддерживающие ООП	18	28
2.	Основные компоненты системы программирования и их роль в разработке программного обеспечения	10	4
3.	Элементы формальных грамматик и теории трансляции.	20	28
	Всего	48	60

Общая трудоемкость дисциплины составляет 3 зачетных единицы (108 часов).

Лекции – 48 часов, самостоятельная работа - 60 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 (по неделям семестра) Форма промежуточной аттестации (по семестрам)
			Лекции	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. Виды отношений между классами. ER-диаграммы.	1	4	4	Самостоятельная работа
1.2	Перегрузка функций. Алгоритм выбора перегруженной функции.	2	2	4	Самостоятельная работа
1.3	Множественное наследование. Исключения в С++. Динамическая идентификация типа.	3	4	6	Самостоятельная работа
1.4	Полиморфизм в С++. Параметрический полиморфизм. Шаблоны функций и классов .	4	2	4	Самостоятельная работа
1.5	Обзор и перспективы нового стандарта языка С++.	5	2	4	Самостоятельная работа
1.6	Решение типовых задач по курсу.	5	2	6	Самостоятельная работа
1.7	Коллоквиум	6	2	0	Коллоквиум по пройденным темам
2.1	Системы программирования и их роль в разработке программного обеспечения	7	2	0	
2.2	Основные компоненты системы программирования	8	4	2	Самостоятельная работа
2.3	Интегрированная среда	9	2	0	

	разработки. Редакторы текстов. Библиотеки. Редакторы связей. Загрузчики. Средства конфигурирования. Средства отладки и тестирования программ.				
2.4	Типы трансляторов, особенности интерпретаторов и компиляторов. Схемы работы трансляторов.	10	2	2	Самостоятельная работа
3.1	Основные понятия теории формальных языков: определение грамматик и языков, классификация по Хомскому. Проблемы однозначности и эквивалентности грамматик. Алгоритм приведения грамматик	11	4	8	Самостоятельная работа
3.2	Регулярные и автоматные грамматики. Диаграммы состояний. Определение детерминированного и недетерминированного конечного автомата. Построение детерминированного конечного автомата по недетерминированному конечному автомату. Лексический анализ на основе регулярных грамматик. Задачи лексического анализа	12	4	6	Самостоятельная работа
3.3	Метод рекурсивного спуска. Критерий применимости метода рекурсивного спуска. Преобразования грамматик, эквивалентность рассматриваемых преобразований. Задачи и проблемы синтаксического анализа.	13	4	6	Самостоятельная работа,
3.4	Задачи семантического анализа. Грамматики с действиями. Контроль	14	2	2	Самостоятельная работа,

	контекстных условий в выражениях.				
3.5	Внутреннее представление программ, свойства языков внутреннего представления. ПОЛИЗ, алгоритм интерпретации ПОЛИЗ. Синтаксически управляемый перевод.	15	2	2	
3.6	Решение типовых задач по курсу.	15	2	4	Самостоятельная работа,
3.7	Коллоквиум.	16	2	0	Коллоквиум по пройденным темам. Экзамен.

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

Лектор курса периодически проводит консультации по дисциплине.

4.6. Интерактивные образовательные технологии, используемые в аудиторных занятиях

Используются традиционные технологии проведения лекций в аудиториях, а также чтение лекций с использованием слайдов. Все методические материалы для прохождения дисциплины отражены на сайте в Интернете.

5. Содержание дисциплины «Системы программирования».

Содержание лекций

Лекция №1

Обзор основных свойств и возможностей объектно-ориентированного программирования на языке Си++, рассмотренных в курсе “ООП” в 3 семестре.

Лекция №2

Операции **new** и **delete**. Работа с динамической памятью. Создание и уничтожение массивов объектов. Плоские и неплоские классы. Примеры.

Виды отношений между классами. ER-диаграммы. Ассоциации классов. Взаимодействие и иерархия классов. (Основные понятия модели Entity-Relationship. Ассоциация классов, агрегация классов, использование одним классом другого класса, инстанцирование класса, наследование одним классом свойств другого класса.)

Лекция №3

Статический полиморфизм в С++. Перегрузка функций. Алгоритм выбора перегруженной функции. Примеры.

Лекция №4

Множественное наследование в С++. Основные проблемы и способы их решения. (Доступ к членам производного класса Преобразование указателей. Виртуальные базовые классы. Неоднозначность из-за совпадающих имён в различных базовых классах.)

Лекция №5

Исключения в С++. Динамическая идентификация типа, средства преобразования типов. Примеры.

Лекция №6

Полиморфизм в C++. Основные виды полиморфизма. Параметрический полиморфизм. Программирование с помощью шаблонов. Шаблоны функций и классов. Перегрузка шаблонных функций.

Лекция №7

Обзор и перспективы нового стандарта языка C++. Примеры.

Лекция №8

Решение типовых задач по курсу. (Решение задач из задачника основной литературы []).

Лекция №9

Коллоквиум.

Лекция №10-12

Определение понятия "Система программирования". Иерархия программно-аппаратного обеспечения. Программа, программный продукт, системный программный продукт. Этапы жизненного цикла программного продукта (фаза разработки, фаза использования и фаза сопровождения).

Технологии разработки программного обеспечения. Общая характеристика этапов разработки и сопровождения программных продуктов. Последовательность этапов разработки во времени.

Компоненты СП, поддерживающие разработчика. Основные требования к системам программирования. Требования к составу систем программирования. Пример системы программирования операционной системы UNIX.

Классическая система программирования. Состав и схема функционирования классической системы программирования. Краткая характеристика отдельных компонентов СП, их роль и положение в общей схеме СП.

Лекция №13

Компоненты классической системы программирования. Интегрированная среда разработки. Редакторы текстов. Библиотеки. Статические и динамически подключаемые библиотеки. Редакторы связей. Загрузчики. Средства конфигурирования. Системы управления версиями программных комплексов. Средства отладки и тестирования программ. Профилировщики. Справочные системы. Краткий обзор современных систем программирования, поддерживающих ООП, и их возможностей.

Лекция №14

Типы трансляторов, особенности интерпретаторов и компиляторов. Схемы работы трансляторов. Интерпретаторы и компиляторы. Смешанная стратегия трансляции. Общая схема работы компилятора. Основные этапы компиляции. Однопроходный компилятор.

Лекция №15-16

Основные понятия теории формальных грамматик. Определения алфавита, цепочки, языка. Определение порождающей грамматики. Выводимость цепочек. Определение языка, порождаемого грамматикой. Эквивалентность грамматик. Определение типов грамматик и языков по Хомскому. Соотношения между типами грамматик. Соотношения между типами языков. Примеры грамматик и языков. Вывод. Цепочки вывода. Сентенциальные формы. Дерево вывода. Неоднозначность вывода и грамматики. Неоднозначность языка. Преобразования грамматик. Определение недостижимых и бесполезных символов, определение приведённой грамматики.

Лекция №17

Регулярные и автоматные грамматики. Леголинейные и праволинейные регулярные грамматики. Алгоритм получения леголинейной грамматики по праволинейной. Диаграммы состояний. Построение диаграммы состояний. Определение детерминированного и недетерминированного конечного автомата. Построение детерминированного конечного автомата по недетерминированному конечному автомату.

Лекция №18

Лексический анализ на основе регулярных грамматик. Задачи лексического анализа. Иерархия классов лексического анализатора модельного языка. Схема работы лексического анализатора модельного языка.

Лекция №19

Метод рекурсивного спуска: назначение, семантика процедур метода рекурсивного спуска. Достаточные условия применимости метода рекурсивного спуска. Подкласс грамматик, к которому применим метод. Критерий применимости метода рекурсивного спуска.

Лекция №20

Задачи и проблемы синтаксического анализа. Расширение класса грамматик, к которым применим метод рекурсивного спуска. Преобразования грамматик, эквивалентность рассматриваемых преобразований. О применимости метода рекурсивного спуска к модельному языку.

Лекция №21

Задачи семантического анализа. Грамматики с действиями. Контроль контекстных условий в выражениях.

Внутреннее представление программ, свойства языков внутреннего представления. ПОЛИЗ выражений, алгоритм интерпретации ПОЛИЗ. Алгоритм Дейкстры. ПОЛИЗ операторов языков программирования. Оператор присваивания. Оператор перехода. Условные операторы и операторы цикла. Операторы ввода и вывода модельного языка. Синтаксически управляемый перевод.

Лекция №22

Синтаксический и семантический анализатор модельного языка. Генерация ПОЛИЗ программы на модельном языке. Интерпретация ПОЛИЗ программы на модельном языке. Генерация кода.

Лекция №23

Решение типовых задач по курсу. (Решение задач из задачника основной литературы []).

Лекция №24

Коллоквиум.

6. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации.

6.1 Коллоквиум № 1. (образец)

NB! Во всех задачах, где это требуется, предполагается наличие необходимых включений и директивы `using namespace std.`

1. При компиляции функции `main ()` следующей программы фиксируется ошибка. Объясните, в чем она заключается и исправьте ее, модифицируя только структуру `B`, ничего в ней не удаляя.

```
struct A {
    int x,y;
    A (int m, int n){ x = m; y = n;}
};
struct B {A a;};

int main () {
    B b;
    cout << b.a.y << '\n';
    return 0;
}
```

2. Вычеркните неразрешимые вызовы функции `f(...)`. Что напечатает получившаяся программа?

```
struct A {
    operator int () { return 1; }
};
void f(double d, char c) { cout << "f(double, char)\n"; }
void f(double d, int j) { cout << "f(double, int)\n"; }
void f(A a, const char * p) { cout << "f(A, const char *)\n"; }
void f(int i, const char * p) { cout << "f(int, const char *)\n"; }

int main () {
    A a;
    f(a, 0);
    f(a, 'a');
    f('a', 0);
    return 0; }
```

3. C++11. Каково назначение служебного слова ***auto*** в C++11? Привести пример его использования.
4. Описать **класс** `B` и необходимые функции так, чтобы при компиляции заданной функции `main ()` в ней не были обнаружены ошибки.

```
int main () {
    B b (1);
}
```

```

        cout << (b++) ++ << endl;
        return 0;
    }

```

5. Что напечатает программа?

```

struct B {
    virtual void f (int a = 0) { cout << a << "B::f (int) \n"; }
    void g () { f (); cout << "B::g () \n"; }
};
class D: public B {
    virtual void f () { cout << "D::f()"; }
    virtual void g () { f (); cout << "D::g()\n"; }
};
int main () {
    D d;
    B b, &rb = d;
    b.g ();
    rb.g (); return 0; }

```

6. Что такое функция-друг в языке Си++? Привести пример описания и использования функции-друга.

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

```

struct A {
    A () { cout << "A_Constr\n"; }
    A (const A & a) { cout << "A_Copy\n"; }
    ~A () { cout << "A_Destr\n"; }
};
struct B : A {
    B () { cout << "B_Constr\n"; }
    B (const B & a) { cout << "B_Copy\n"; }
    ~B () { cout << "B_Destr\n"; }
};

void g() { B bg; throw bg;}

int main () {
    try { A a; g(); }
    catch ( A& ) { cout << "A&_Catch\n"; }
    catch ( B& ) { cout << "B&_Catch\n"; }
    return 0;
}

```

8. Исправьте ошибки в нижеследующей программе, ничего в ней не удаляя.

Объясните, в чем заключаются исправленные ошибки.

Что напечатает программа после внесенных исправлений?

```

struct A {
    int y;
    int f () { return (y = 2); }
};
struct B : A {
    static int g () { return 9 + f(); }
};
int A::y;

int main () {
    B b,d;
    b.y = b.f();
    cout << B::f() << ' ' << B::y << ' ' << d.y <<
B::g() << endl;
    return 0; }

```

9. Опишите структуру с именем S, удовлетворяющую двум условиям:

-- можно создать объект типа S

-- нельзя создать массив элементов типа S в динамической памяти.

10. Модифицируйте программу, **ничего в ней не удаляя**, так, чтобы операция **dynamic_cast** работала безошибочно, а программа в целом завершилась нормально (не аварийно).

```

struct B {
    void g1 () { cout << "B::g1()\n"; }
};
struct D:B {};

int main () {
    D d, * pd;
    B b, * pb = &b, * pbd = &d;
    pd = dynamic_cast <D*> (pbd); pd -> g1();
    pd = dynamic_cast <D*> (pb); pd -> g1();
    return 0; }

```

6.2 Коллоквиум № 2. (образец)

1. Дана грамматика G :

$$S \rightarrow aA$$

$$aA \rightarrow aaB/a$$

$$B \rightarrow abb/aCbb/b$$

$$C \rightarrow ab/aaD$$

$$D \rightarrow bb/abbb$$

(а) Описать язык $L(G)$ в виде теоретико-множественной формулы.

(б) Каким из перечисленных классов грамматик принадлежит G ?

Ответ:

Класс Π	$G \in \Pi?$ (да/нет)
Регулярные	
контекстно-зависимые	
контекстно-свободные	
грамматики типа 0	
неукорачивающие	

(в) Тип языка: найти такое целое k , что язык $L(G)$ является языком типа k и не является языком $k+1$.

2. Дать определение системы программирования.

Из перечисленных компонентов вычеркните те, которые не входят в состав системы программирования:

Текстовый редактор,

Макрогенератор,

Драйвер клавиатуры,

Редактор связей.

3. а) Построить по заданной грамматике G конечный автомат (в виде ДС).

б) Если автомат оказался недетерминированным, преобразовать его в соответствии с алгоритмом преобразования НКА в эквивалентный ДКА.

в) По ДКА построить праволинейную грамматику.

$$G: \quad S \rightarrow C\perp$$

$$C \rightarrow AI$$

$$B \rightarrow BI / 0$$

$$A \rightarrow BI / CO / 0$$

4. Построить неукорачивающую КС-грамматику, эквивалентную данной КС грамматике, пользуясь алгоритмом преобразования «КС- \rightarrow неукорачивающая КС».

$$S \rightarrow ABc / c/A$$

$$A \rightarrow aA / \varepsilon$$

$$B \rightarrow bB / \varepsilon$$

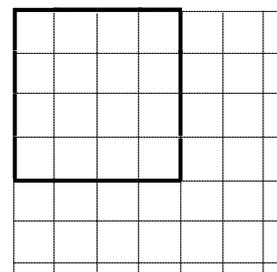
5. Имеется клетчатый лист бумаги, бесконечный вправо и вниз. В левом верхнем углу расположено перо, оставляющее на бумаге след при перемещении. Перо управляется интерпретатором-графопостроителем. На вход интерпретатору подается последовательность символов-команд из алфавита $\{a, b, c, d\}$:

a — переместить перо на одну клетку вправо;

b — вниз; c — влево;

d — вверх.

Например, по последовательности $aaaaabbbbcccc$ интерпретатор построит квадрат, изображенный на рисунке, со стороной длины 4.



Построить грамматику с терминальным алфавитом $\{a, b, c, d\}$, описывающую все квадраты со стороной длины n , $n \geq 1$, начерченные по часовой стрелке. В грамматике должно быть **не более 6** правил вывода (считая альтернативы). Решения с большим числом правил не рассматриваются. Грамматика не должна описывать другие фигуры.

6. 1) Дать определение языка, порождаемого грамматикой $G=(T,N,P,S)$.

2) Привести пример грамматики (задав все элементы порождающей грамматики), которая порождает пустой язык.

7. Применим ли метод рекурсивного спуска к данной грамматике. **Ответ обосновать.**

$$S \rightarrow ABc / c$$

$$A \rightarrow aA / \varepsilon$$

$$B \rightarrow bB / \varepsilon$$

8. Дана польская инверсная запись фрагмента программы на языке Си. Вставьте пропущенные в польской записи команды перехода ('!', '!F') и недостающие метки и восстановите фрагмент на языке Си, если известно, что в исходном фрагменте нет операторов перехода.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
X	0	>			y	0	<			y	y	1	x	+	+	=	;			27		y
24	25	26	27	28																		
X	=	;		{																		

7. *Оценочные средства рубежного контроля*

7.1 *Вопросы к экзамену*

1. Основные понятия и определения теории формальных языков. Примеры.
2. Классификация формальных грамматик и языков по Хомскому. Примеры.
3. Соотношения между типами грамматик, соотношения между типами языков. Эквивалентные и почти эквивалентные грамматики.
4. Бесплодные и недостижимые символы. Приведенные грамматики. Алгоритм приведения грамматики.
5. Вывод. Дерево вывода. Примеры.
6. Понятие неоднозначности грамматики и языка. Примеры.
7. Деревья вывода при разборе по регулярным грамматикам.
8. Регулярные грамматики и конечные автоматы. Алгоритмы построения грамматики по автомату и автомата по леворегулярной грамматике.
9. Детерминированные и недетерминированные конечные автоматы. Алгоритм преобразования НКА в ДКА.
10. Синтаксический анализ. Метод рекурсивного спуска.
11. Модификация метода рекурсивного спуска для грамматик с итерациями и для грамматик, содержащих ϵ -правила. Применимость метода к таким грамматикам.
12. Пути преобразования грамматик, не удовлетворяющих условиям применимости метода рекурсивного спуска.
13. Генерация кода и синтаксически управляемый перевод. примеры.
14. Польская инверсная запись (ПОЛИЗ) арифметических выражений. Интерпретация ПОЛИЗа. Алгоритм Дейкстры для перевода выражений в ПОЛИЗ.
15. Расширение ПОЛИЗа для представления произвольных программ; ПОЛИЗ как язык внутреннего представления программ.
16. Жизненный цикл программного продукта.
17. Понятие и состав системы программирования, её роль и место в составе вычислительной системы
18. Требования к редактору текстов, предназначенному для программирования.
19. Понятия интерпретатора и компилятора, схемы их работы. Смешанная стратегия трансляции.

20. Объектный код, объектные модули и редактор связей. Библиотеки.
21. Программное обеспечение для поддержки отладки программ. Примеры.
22. Объектно-ориентированное программирование (ООП) -- новая технология (парадигма) программирования. ООП-анализ.
23. Пространства имен в языке Си++.
24. Основные свойства языка, поддерживающего ООП.
25. Понятие класса и объекта. Описание класса.
26. Управление доступом к членам класса -- public, private, protected.
27. Операции . и ->, символ ::, указатель this.
28. Объявления и описания функций-членов класса; эффект inline.
29. Специальные функции -- конструкторы и деструктор.
30. Конструктор копирования.
31. Конструктор копирования и операция присваивания: содержательная связь и различие.
32. Ссылки и указатели в Си++.
33. Операторы new и delete.
34. Друзья класса, "законы" дружбы. Сравнение функции-члена и функции-друга: описание, вызов.
35. Обработка исключений в С++. Преобразование типов в обработчиках исключений.
36. Перегрузка функций. Перегрузка и неоднозначность.
37. Функции с параметрами по умолчанию.
38. Алгоритм поиска оптимально отождествляемой (best-matching) функции (1-3 шага - стандартные преобразования).
39. Алгоритм поиска оптимально отождествляемой (best-matching) функции (4-5 шага - пользовательские преобразования).
40. Перегрузка операторов. Перегрузка с помощью функции-члена и функции-друга.
41. Перегрузка бинарных операций в С++.
42. Перегрузка унарных операций в С++.
43. Особенности перегрузки операций ++ и --, операции индексации в С++.
44. Особенности перегрузки операции присваивания.
45. Статические члены класса.
46. Константные функции-члены.
47. Одиночное наследование. Правила наследования. Видимость при наследовании.
48. Конструкторы и деструкторы при наследовании.
49. Указатели на базовый и производный классы, преобразование указателей.

50. Динамический полиморфизм. Виртуальные функции.

51. Чисто виртуальные функции. Абстрактные классы.

52. Динамическая информация о типе (RTTI).

53. Параметрический полиморфизм. Шаблонные функции.

54. Шаблонные классы.

8. Учебно-методическое и информационное обеспечение дисциплины (модуля)

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. 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>
3. И. А. Волкова, И. Г. Головин, Л. Е. Карпов. Системы программирования (Учебное пособие) . — М.: Издательский отдел факультета ВМиК МГУ, 2009.
Электронная версия:
<http://cmcmsu.no-ip.info/download/programming.systems.course.pdf>
4. 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
5. 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>

б) дополнительная литература:

1. Б. Страуструп. Язык программирования С++. Специальное издание. — М.: Издательство «БИНОМ», 2001. (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. Дизайн и эволюция С++. Пер. с англ. – М.: ДМК Пресс, 2000.
4. Д. Грис. Конструирование компиляторов для цифровых вычислительных машин. — М.: Мир, 1975.

5. 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>
6. А. Ахо, Дж. Ульман. Теория синтаксического анализа, перевода и компиляции, т.1,2 — М.: Мир, 1979.
7. А. Ахо, Р. Сети, Дж. Ульман. Компиляторы. — М.: Изд. дом «Вильямс», 2001. (Шифр в библиотеке МГУ: 5ВГ66 А-955)
8. Страуструп Б. Программирование: принципы и практика использования С++.: Пер. с англ. – М. ООО «И.Д.Вильямс», 2011.
9. Пол А. "Объектно-ориентированное программирование на Си++" – второе издание, М., Бином, 1999.
10. . В. Гордеев, А. Ю. Молчанов. Системное программное обеспечение. — СПб.: Питер, 2001
11. Г. Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е издание. — М. СПб.: «Издательство Бином» — «Невский диалект»,1998.
12. И. О. Одинцов. Профессиональное программирование. Системный подход. — СПб.: БХВ-Петербург, 2002. (Шифр в библиотеке МГУ: 5ВГ66 О-425)
13. Г. Майерс. Искусство тестирования программ. — М.: «Финансы и статистика», 1982

в) Программное обеспечение и Интернет-ресурсы. Студентам предлагается искать дополнительную информацию на сайтах, посвященных системам программирования.

9. Материально-техническое обеспечение дисциплины (модуля)

Наличие литературы в библиотеке, медиапроектор и компьютер для проведения лекций-презентаций.

Программа составлена в соответствии с требованиями ФГОС ВПО, ОС МГУ

«Прикладная математика и информатика», с учетом рекомендаций Примерной основной образовательной программы (ПрООП) по направлению «010300 Фундаментальные информатика и информационные технологии», бакалавриат.