

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
«Московский физико-технический институт (государственный университет)»
МФТИ

«УТВЕРЖДАЮ»
Проректор по учебной и методической работе
_____ Д.А. Зубцов
« » _____ 20 г.

Рабочая программа дисциплины (модуля)

по дисциплине: Вычислительные средства: C++

по направлению: 010900 «Прикладные математика и физика»

профиль подготовки

магистерская программа: 010915 «Физика высоких энергий»

факультет: ФОПФ

кафедра: Фундаментальных и прикладных проблем физики микромира

курс: 4

квалификация: бакалавр

Семестр, формы промежуточной аттестации: 8 (Весенний) Диф.зачет

Аудиторных часов: 102 всего, в том числе:

лекции: 34 час.

практические (семинарские) занятия: 68 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 34 час., в том числе:

задания, курсовые работы: 0 час.

Подготовка к экзамену: 0 час.

Всего часов: 136, всего зач. ед.: 4

Программу составил: к.ф.-м.н. Нефёдов Ю.А.

Программа обсуждена на заседании кафедры

14 октября 2014 г.

СОГЛАСОВАНО:

Заведующий кафедрой

Казаков Д.И.

Декан ФОПФ

Трунин М.Р.

Начальник учебного управления

Гарайшина И.Р.

1. Цели и задачи

Цель дисциплины

В современном мире программирование является рутинным занятием ученого, будь то обработка экспериментальных данных или теоретические расчеты. В физике и особенно в физике высоких энергий возросшая сложность экспериментов привела к доминированию программного обеспечения написанного на языке C++. Язык C++, обладая несомненной гибкостью и мощностью заложенных в него концепций, сложен в изучении, особенно для начинающих. В данном курсе лекций дается базовое описание языка C++ и студенты знакомятся с основами объектно-ориентированного программирования. Семинарские занятия строятся в параллель с лекциями и содержат решение типичных задач иллюстрирующих изучаемые на лекциях темы

Задачи дисциплины

- формирование базовых знаний языка программирования C++;
- знакомство студентов с основами методик императивного и объектно-ориентированного программирования;
- формирование у студентов практических навыков разработки компьютерных программ.

2. Место дисциплины (модуля) в структуре образовательной программы бакалавриата

Дисциплина «Вычислительные средства: C++» включает в себя разделы, которые могут быть отнесены к вариативной части профессионального цикла ООП Б.3.

Дисциплина «Вычислительные средства: C++» базируется на материалах курсов, читаемых в рамках базовой и вариативной частей УЦ ООП Б.2 и Б.3 (Введение в программирование, Объектно-ориентированное программирование, Алгоритмы и структуры данных), и относится к профессиональному циклу.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Освоение дисциплины «Вычислительные средства: C++» направлено на формирование следующих общекультурных и профессиональных компетенций бакалавра:

а) общекультурные (ОК):

- владение культурой мышления, способностью к обобщению, анализу, восприятию информации, постановке целей и выбору путей её достижения (ОК-1);
- способность к саморазвитию, повышению квалификации, устранению пробелов в знаниях и самостоятельному обучению в контексте непрерывного образования, способность осваивать новую проблематику, язык, методологию и научные знания в избранной предметной области (ОК-6);
- способность к применению основных методов, способов и средств получения, хранения, переработки информации, к работе с компьютером как средством управления информацией (ОК-11);

- способность работать с информацией в глобальных компьютерных сетях (ОК-12);

б) профессиональные (ПК):

- способность формализовать и решать отдельные части нестандартной задачи в общей постановке (ПК-1);
- способность к пониманию важности воздействия внешних факторов, и их учёта в ходе исследований и разработок (ПК-2);
- способность применять основные законы естественнонаучных дисциплин в профессиональной деятельности, применять методы математического анализа и моделирования, теоретического и экспериментального исследования в физике, химии, экологии, других естественных и социально-экономических науках (ПК-3);
- способность к выявлению сущности задач, возникающих в ходе профессиональной деятельности, и привлечению соответствующего физико-математического аппарата для их решения (ПК-4);
- способность самостоятельно работать на компьютере на уровне квалифицированного пользователя, применять информационно-коммуникационные технологии для обработки, хранения, представления и передачи информации с использованием универсальных пакетов прикладных программ, знание общих подходов и методов по совершенствованию информационно-коммуникационных технологий (ПК-6);
- способность брать на себя ответственность за качество и результаты своей деятельности (ПК-10).

В результате освоения дисциплины «Вычислительные средства: C++» обучающийся должен:

знать:

- основы языка программирования C++;
- типовые решения, применяемые при написании программ на C++ ;
- принципы программирования структур данных;
- основы парадигм императивного и объектно-ориентированного программирования.

уметь:

- работать в операционной системе UNIX: редактировать файлы (emacs), компилировать и отлаживать программы (gcc), пользоваться базовыми командами командного интерпретатора (bash);
- применять объектно-ориентированный подход для написания программ.

владеть:

- языком программирования C++;

- средствами программирования базовых структур данных (vector, list и map) с использованием стандартной библиотеки шаблонов (STL).

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№ п/п	Тема (раздел) дисциплины	Виды учебных занятий, включая самостоятельную работу				
		Лекции	Практич. (семинар.) задания.	Лаборат. работы	Задания, курсовые работы	Самост. работа.
1	Введение.	2	0	0	0	2
2	От С к С++	2	2	0	0	2
3	Ссылки (references).	2	4	0	0	2
4	Пользовательские типы данных: классы.	2	4	0	0	2
5	Конструкторы и деструкторы.	2	4	0	0	2
6	Перегрузка операторов и функций.	2	4	0	0	2
7	Наследование.	2	4	0	0	2
8	Наследование и полиморфизм.	2	4	0	0	2
9	Исключения (exceptions).	2	4	0	0	2
10	С++ стиль ввода-вывода.	2	4	0	0	2
11	Шаблоны (templates).	2	6	0	0	2
12	STL: Последовательный контейнер vector.	2	4	0	0	2
13	STL: Последовательный контейнер list.	2	6	0	0	2
14	STL: Ассоциативные контейнеры set и map.	2	6	0	0	2
15	STL: Алгоритмы.	4	8	0	0	2
16	Стандартная библиотека С++: string, complex.	2	4	0	0	4
Итого часов		34	68	0	0	34
Общая трудоемкость		102 час., 4 зач.ед.				

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 8 (Весенний)

1 Введение.

Эволюция языков программирования. Программа "HELLO, WORLD" в C++. Транслятор языка C++. Рекомендуемая литература.

2 От C к C++

C++ как «расширенный» C. Новый стиль ввода-вывода. Пространства имён. Перегрузка функций. Прототипы и сигнатуры.

3 Ссылки (references).

Ссылки (references). Динамическая память, C++-стиль.

4 Пользовательские типы данных: классы.

Пользовательские типы данных: классы. Зачем нужны классы если уже есть структуры? Объектный стиль программирования.

5 Конструкторы и деструкторы.

Конструкторы и деструкторы. Конструктор по умолчанию. Копирующий конструктор. Конструкторы в качестве операторов приведения типа.

6 Перегрузка операторов и функций.

Перегрузка операторов и функций. Друзья классов.

7 Наследование.

Наследование. Виртуальные функции. Абстрактные классы.

8 Наследование и полиморфизм.

Наследование и полиморфизм. Основная идея. RTTI и `dynamic_cast`.

9 Исключения (exceptions).

Исключения (exceptions). Исключения и полиморфизм. Перехват исключений оператора `new`.

10 C++ стиль ввода-вывода.

C++ стиль ввода-вывода. Перегрузка операторов ввода-вывода. Управление форматным выводом. Манипуляторы.

11 Шаблоны (templates).

Понятие обобщённого (generic) программирования и введение в стандартную библиотеку шаблонов (STL). Рекомендуемая литература по STL. Программирование простейших шаблонов функций.

12 STL: Последовательный контейнер vector.

Использование STL: Контейнеры, алгоритмы, итераторы. Последовательный контейнер vector.

13 STL: Последовательный контейнер list.

Использование STL: Последовательный контейнер list.

14 STL: Ассоциативные контейнеры set и map.

Использование STL: Ассоциативные контейнеры set и map.

15 STL: Алгоритмы.

Использование STL: Алгоритмы. Объекты-функции (function object).

16 Стандартная библиотека C++: string, complex.

Стандартная библиотека C++: string, complex. Полезные функции. C++, что осталось «за кадром»?

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Необходимое оборудование для лекций и практических занятий: компьютер и мультимедийное оборудование (проектор). Практические занятия проводятся в компьютерном классе.

Необходимое программное обеспечение: набор компиляторов GCC.

Обеспечение самостоятельной работы: доступ к программному обеспечению с установленными компиляторами GCC.

6. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

Основная литература:

- 1 Б. Страуструп «Язык программирования C++»
- 2 Н. Джосьютис «C++. Стандартная библиотека»

7. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

-

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Информационные ресурсы:

- <http://www.parashift.com/c++-faq-lite/> – Ответы на часто задаваемые вопросы по C++
- <http://www.sgi.com/tech/stl/> – Руководство программиста «Стандартная библиотека шаблонов»

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

-

10. Методические указания для обучающихся по освоению дисциплины

-

11. Фонд оценочных средств для проведения промежуточной аттестации по итогам обучения

Перечень контрольных заданий для сдачи дифференциального зачета в 8-ом семестре:

1) Запрограммировать класс для работы с рациональными числами. Класс должен содержать:

- Конструкторы для инициализации как целым числом так и числителем и знаменателем.
- Операторы +, -, *, /, +=, -=, *=, /=, префиксные и постфиксные ++ и --
- Перегрузить функцию возведения в степень: pow().
- Оператор вывода рационального числа на печать.
- Преобразование к несократимому виду так что бы числитель и знаменатель были взаимно просты (Алгоритм Евклида).
- Выполните тестирование всех операций и функций на простых примерах.

2) Запрограммировать класс для многочленов. Коэффициенты многочлена хранить в контейнере vector<double>. Класс должен содержать:

- Оператор вывода на печать
- Оператор () для вычисления многочлена по схеме Горнера
- Операторы сложения (вычитания) и умножения
- Функцию вычисляющую производную от многочлена
- Функция для деления многочленов «столбиком»

3) Запрограммировать простой телефонный справочник (имя -> номер телефона) на основе контейнера map. Реализовать:

- Функцию добавление новой записи
- Функцию поиска по имени
- Функцию удаления записи

Написать программу тестирующую все эти функции