

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

Саровский физико-технический институт -

филиал федерального государственного автономного образовательного учреждения высшего
образования «Национальный исследовательский ядерный университет «МИФИ»
(СарФТИ НИЯУ МИФИ)

ФИЗИКО-ТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра «Прикладной математики»

УТВЕРЖДАЮ

Декан ФТФ, член корр. РАН, д.ф-м.н.

_____ **А.К. Чернышев**

« ____ » _____ **2023 г.**

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Языки и методы программирования

наименование дисциплины

Направление подготовки (специальность)	<u>01.03.02 Прикладная математика и информатика</u>
Наименование образовательной программы	<u>Высокопроизводительные вычисления и технологии параллельного программирования</u>
Квалификация (степень) выпускника	<u>бакалавр</u>
Форма обучения	<u>очная</u>

Программа одобрена на заседании кафедры

Зав. кафедрой ПМ, д.ф-м.н.

_____ **Р.М. Шагалиев**

протокол № от

20 г.

« ____ » _____ **2023 г.**

г. Саров, 2023 г.

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФИТЭ, ФТФ на 202____/202____ учебный год.

Заведующий кафедрой ПМ, д.ф.-м.н.

Р.М. Шагалиев

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФИТЭ, ФТФ на 202____/202____ учебный год.

Заведующий кафедрой ПМ, д.ф.-м.н.

Р.М. Шагалиев

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФИТЭ, ФТФ на 202____/202____ учебный год.

Заведующий кафедрой ПМ, д.ф.-м.н.

Р.М. Шагалиев

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФИТЭ, ФТФ на 202____/202____ учебный год.

Заведующий кафедрой ПМ, д.ф.-м.н.

Р.М. Шагалиев

Семестр	В форме практической подготовки	Трудоемкость, кред.	Общий объем курса, час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	КР/ КП	Форма(ы) контроля, экз./зач./ЗсО/	Интерактивные часы
1	32	4	144	16	-	32	60	-	Э 36	8
2	32	3	108	16	-	32	33	-	Э 27	8
ИТОГО	64	7	252	32	-	64	93	-	63	16

АННОТАЦИЯ

В рамках курса студенты должны разобраться в вычислительных процессах и алгоритмизации процессов обработки данных; в языках программирования и их назначении.

В среде программирования должны усвоить представление данных в языке программирования; выражения, встроенные функции и процедуры; описание базовых структур; программирование линейных и ветвящихся вычислительных процессов; процедуры и их использование в программах; организация ввода и вывода данных; программирование и построение и ввода табличных форм; работа с файлами; динамические переменные и указатели.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В дисциплинах «Языки и методы программирования» рассматриваются теоретические и практические аспекты основ алгоритмизации и программирования.

Изучаются способы и методы разработки эффективных алгоритмов для решения различных прикладных задач. Рассматриваются общие принципы создания программ на языках высокого уровня и изучаются языки программирования C, C++, Visual C++, Qt.

Главная цель преподавания дисциплин – подготовка специалиста, владеющего фундаментальными знаниями и практическими навыками в области основ алгоритмизации и программирования.

Задачи дисциплин - дать основы:

- ✓ Понятий алгоритмов, их классификаций, методов описания и представления алгоритмов, предназначенных для написания программ, используемых для решения задач из различных предметных областей;
- ✓ Структур языков программирования и инструментальных сред разработки программных продуктов;

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплины «Языки и методы программирования» являются базовыми дисциплинами образовательного стандарта высшего профессионального образования (ОС ВО) по направлению «Прикладная математика и информатика» (квалификация – «бакалавр»).

Изучение дисциплины «Языки и методы программирования» основывается на базе знаний, умений и компетенций, полученных студентами в ходе освоения школьного курса «Информатика».

Дисциплины «Языки и методы программирования» являются базовым теоретическим и практическим основанием всех последующих дисциплин подготовки бакалавра.

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Ожидается, что в результате освоения дисциплины студент приобретет следующие компетенции:

Общепрофессиональные компетенции (ОПК)

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности	З-ОПК-1 знать естественнонаучные методы познания окружающего мира, знать фундаментальный математический аппарат; У-ОПК-1 уметь применять естественнонаучные и математические методы исследования различных явлений, процессов и задач В-ОПК-1 владеть навыками исследования различных явлений и процессов с использованием естественнонаучного и математического подхода
ОПК-3 Способен применять и модифицировать математические модели для решения задач в области профессиональной деятельности	З-ОПК-3 знать принципы построения математических моделей физических явлений и процессов У-ОПК-3 уметь формулировать математические модели различных явлений и процессов на основе физических принципов и законов В-ОПК-3 владеть навыками построения математических моделей физических явлений и процессов
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	З-ОПК-5 Знать основные языки программирования и методы алгоритмизации, современные технические и программные средства для разработки компьютерных программ У-ОПК-5 Уметь применять методы алгоритмизации и современные технологии программирования для решения практических задач в различных областях науки и техники В-ОПК-5 Владеть навыками разработки алгоритмов и компьютерных программ, отладки и тестирования разработанных программных комплексов для решения научно-практических задач

4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ*

№ п/п	Наименование раздела /темы дисциплины	№ недели	Виды учебной работы					
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			32	-	64	93		
Семестр 1								
Раздел 1. Основы алгоритмизации. Основы структурного программирования Системы программирования								
1.1	Тема 1. Алгоритм и его свойства.	1	2				УО	
1.2	Тема 2. Теория структурного программирования.	1			4	4	УО	
1.3	Тема 3. Назначение и состав системы программирования.	2			4	8	УО	5
Раздел 2. Язык программирования С. Элементы языка. Данные. Константы и переменные. Выражения								
2.1	Тема 1. Этапы развития языка.	3	2				УО	
2.2	Тема 2. Стандарты.	3			4	8	УО	
2.3	Тема 3. Данные в языке С.	4	2				УО	
2.4	Тема 4. Операции и их классификация, приоритеты операций.	4			4	8	УО	5
Раздел 3. Операторы управления. Функции. Сложные типы данных. Динамическая память. Операторы ввода-вывода								
3.1	Тема 1. Общая характеристика и назначение операторов управления.	5	2				УО	
3.2	Тема 2. Общее определение и применение функции как самостоятельной программной единицы.	6			4	8	УО	
3.3	Тема 3. Массивы: одномерные (векторы), двумерные (матрицы), многомерные массивы. Классы памяти.	7	2				УО	
3.4	Тема 4. Общие принципы организации систем ввода-вывода.	8			4	8	УО	10
Рубежный контроль		8	ИЗ					10
Раздел 4. Основные понятия ООП. Конструкторы и деструкторы. Дружественные функции и дружественные классы. Наследование классов								
4.1	Тема 1. Важнейшие характеристики ООП.	9	2				УО	
4.2	Тема 2. Понятие и роль функции в классе. Функции конструкторы и деструкторы их назначение и типы.	10	2				УО	

№ п/п	Наименование раздела /темы дисциплины	№ недели	Виды учебной работы					
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			32	-	64	93		
4.3	Тема 3. Понятие дружественной функции и дружественного класса. Общее понятие наследования классов.	11-12			4	8	УО	
Раздел 5. Перегрузка функций и операций. Перегрузка операций ввода-вывода. Ссылки								
5.1	Тема 1. Общее понятие перегрузки функций. Общие принципы перегрузки операций ввода-вывода в рамках класса.	13-14	2				УО	
5.2	Тема 2. Ссылки и их роль и назначение при передаче параметров функций.	15-16			4	8	УО	10
Рубежный контроль		16						ИЗ
Промежуточная аттестация							Экзамен	36
Посещаемость								
Итого:			16	-	32	60	36	100
Семестр 2								
Раздел 6. Виртуальные функции и виртуальные классы. Шаблоны функций и шаблоны классов. Статические, локальные и вложенные классы.								
6.1	Тема 1. Полиморфизм в C++. Базовые классы и виртуальные функции – механизм реализации полиморфизма.	1	2				УО	
6.2	Тема 2. Особенности использования виртуальных функций.	2			4	4	УО	5
6.3	Тема 3. Шаблоны функций для реализации полиморфизма в C++.	3	2				УО	
6.4	Тема 4. Статические данные и статические функции класса.	4			4	4	УО	5
Раздел 7. Использование класса QString для создания класса по работе с большими числами. Контейнерные классы. Обработка массивов с использованием класса QVector								
7.1	Тема 1. Пример класса для работы с большими числами.	5	2				УО	
7.2	Тема 2. Краткое описание контейнерных классов.	6			4	4	УО	5
7.3	Тема 3. Пример использования класса QVector для решения задач по обработке массивов.	7	2				УО	
7.4	Тема 4. Вывод массивов на экран.	8			4	4	УО	5

№ п/п	Наименование раздела /темы дисциплины	№ недели	Виды учебной работы					
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			32	-	64	93		
Рубежный контроль		8	ИЗ					10
Раздел 8. Создание и обработка таблиц. Клиент-серверные приложения (консольные). Клиент-серверные приложения (с графическим интерфейсом)								
8.1	Тема 1. Класс QTableWidgetItem.	9	2				УО	
8.2	Тема 2. Основные сведения об организации передачи информации по сети.	10			4	4	УО	
8.3	Тема 3. Класс QSocet.	11	2				УО	
8.4	Тема 4. Пример организации клиент-серверного приложения с графическим интерфейсом.	12			4	4	УО	5
Раздел 9. Поток. Построение графиков. Имитация движения на экране								
9.1	Тема 1. Процессы и потоки.	13	2				УО	
9.2	Тема 2. Построение графиков функций.	14			4	4	УО	5
9.3	Тема 3. Класс QTimer.	15	2				УО	
9.4	Тема 4. Смена изображений на экране по сигналам таймера.	16			4	5	УО	
Рубежный контроль		16	ИЗ					10
Промежуточная аттестация			Экзамен				27	45
Посещаемость								5
Итого:			16	-	32	33	27	100

*Сокращение наименований форм текущего, рубежного и промежуточного контроля:

УО – устный опрос

ИЗ – индивидуальное задание

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс

№	Наименование раздела /темы дисциплины	Содержание
1 семестр		
Раздел 1. Основы алгоритмизации. Основы структурного программирования Системы программирования		
1.1	Тема 1. Алгоритм и его свойства.	Алгоритм и его свойства. Разновидность структур алгоритмов. Способы описания алгоритмов. Стандартизация графического представления алгоритмов.
1.2	Тема 2. Теория структурного программирования.	Теория структурного программирования. Реализация основ структурного программирования в языках программирования. Представление структурированных схем.
1.3	Тема 3. Назначение и состав системы программирования.	Назначение и состав системы программирования. Классификация языков программирования. Инструментальная среда Microsoft Visual Studio 6.0. Система программирования Visual C++ 6.0
Раздел 2. Язык программирования С. Элементы языка. Данные. Константы и переменные. Выражения		
2.1	Тема 1. Этапы развития языка.	Этапы развития языка. Основные элементы и конструкции языка: константы и переменные, ограничители, разделители, ключевые слова, идентификаторы, операции и операторы.
2.2	Тема 2. Стандарты.	Стандарты. Основные элементы и конструкции языка: константы и переменные, ограничители, разделители, ключевые слова, идентификаторы, операции и операторы.
2.3	Тема 3. Данные в языке С.	Данные в языке С. Константы целого, вещественного, символьного и строкового типа. Переменные, память, занимаемая переменными различного типа, диапазоны допустимых значений переменных разного типа.
2.4	Тема 4. Операции и их классификация, приоритеты операций.	Операции и их классификация, приоритеты операций. Выражения и их типы, правила вычисления выражений. Выражения в операторах присваивания, преобразование типов.
Раздел 3. Операторы управления. Функции. Сложные типы данных. Динамическая память. Операторы ввода-вывода		
3.1	Тема 1. Общая характеристика и назначение операторов управления.	Общая характеристика и назначение операторов управления. Операторы условного перехода (операторы if, if else, switch). Операторы цикла (операторы while, do while, for, break, continue). Применение операторов цикла при работе с массивами. Вложенные циклы.
3.2	Тема 2. Общее определение и применение функции как самостоятельной программной единицы.	Общее определение и применение функции как самостоятельной программной единицы. Структурное строение функции. Прототипы функций. Интерфейс вызова функций. Классификация функций: функции общего назначения, арифметические и математические функции, функции работы со строками, системные функции. Общие замечания и рекомендации к разработке и применению функций в программных комплексах.
3.3	Тема 3. Массивы: одномерные (векторы), двумерные (матрицы), многомерные массивы. Классы памяти.	Массивы: одномерные (векторы), двумерные (матрицы), многомерные массивы. Инициализации массивов. Массивы символов, строки. Структуры и объединения. Агрегатные переменные, задаваемые пользователем. Общие замечания и рекомендации к применению сложных переменных. Классы памяти. Статическая и динамическая память. Общие принципы работы с динамической памятью. Функции malloc, free, new, delete, особенности их применения при работе с динамической памятью.
3.4	Тема 4. Общие принципы организации систем ввода-вывода.	Общие принципы организации систем ввода-вывода. Понятия файла, потока. Типы файлов. Классификация функций ввода-вывода. Функции работы с экраном и клавиатурой. Функции работы с текстовыми и бинарными файлами. Обработка исключительных ситуаций при работе с файлами. Особенности применения различных систем ввода-вывода в языке С.

Раздел 4. Основные понятия ООП. Конструкторы и деструкторы. Дружественные функции и дружественные классы. Наследование классов		
4.1	Тема 1. Важнейшие характеристики ООП.	Объектно-ориентированное программирование – новый современный подход к разработке программных продуктов. Язык C++. Основные понятия. Объекты и классы. Важнейшие характеристики ООП: инкапсуляция, наследование, полиморфизм, абстракция типов. Режимы доступа при использовании данных и функций класса. Пример класса – очередь.
4.2	Тема 2. Понятие и роль функции в классе. Функции конструкторы и деструкторы их назначение и типы.	Понятие и роль функции в классе. Функции конструкторы и деструкторы их назначение и типы. Конструктор по умолчанию, основной конструктор, конструктор копирования. Параметры в конструкторах при работе с объектами. Примеры программ с использованием конструкторов и деструкторов.
4.3	Тема 3. Понятие дружественной функции и дружественного класса. Общее понятие наследования классов.	Понятие дружественной функции и дружественного класса. Назначение и использование дружественных функций и дружественных классов в C++. Реализация доступа к данным класса через дружественные функции. Примеры программ с использованием дружественных функций. Общее понятие наследования классов. Атрибуты и свойства объектов при наследовании. Доступ к данным и функциям при наследовании. Простое и множественное наследование. Примеры программ с использованием механизма наследования.
Раздел 5. Перегрузка функций и операций. Перегрузка операций ввода-вывода. Ссылки		
5.1	Тема 1. Общее понятие перегрузки функций. Общие принципы перегрузки операций ввода-вывода в рамках класса.	Общее понятие перегрузки функций. Перегрузка конструкторов. Перегрузка операций в рамках класса. Оператор-функция operator. Механизмы и принципы перегрузки операций. Список операций, допускающих перегрузку. Примеры программ, использующих перегрузку функций и операций. Общие принципы перегрузки операций ввода-вывода в рамках класса. Инсерторы и экстракторы. Особенности применения перегруженных операций ввода-вывода. Примеры программ с перегрузкой операций ввода-вывода.
5.2	Тема 2. Ссылки и их роль и назначение при передаче параметров функций.	Ссылки и их роль и назначение при передаче параметров функций. Использование ссылочных переменных при перегрузке унарных операций и операции индексации. Отличие механизма использования ссылок от передачи параметров по значению и через указатели. Примеры программ с использованием ссылок.
2 семестр		
Раздел 6. Виртуальные функции и виртуальные классы. Шаблоны функций и шаблоны классов. Статические, локальные и вложенные классы.		
6.1	Тема 1. Полиморфизм в C++. Базовые классы и виртуальные функции – механизм реализации полиморфизма.	Полиморфизм в C++. Базовые классы и виртуальные функции – механизм реализации полиморфизма.
6.2	Тема 2. Особенности использования виртуальных функций.	Особенности использования виртуальных функций. Чистые виртуальные функции. Виртуальные классы и их применение. Примеры программ с применением виртуальных функций и виртуальных классов.
6.3	Тема 3. Шаблоны функций для реализации полиморфизма в C++.	Шаблоны функций для реализации полиморфизма в C++. Использование типов переменных в качестве параметров. Построение отдельных классов посредством шаблонов классов. Передача параметризованных типов при использовании шаблонов классов. Примеры программ с использованием шаблонов функций и шаблонов классов.
6.4	Тема 4. Статические данные и статические функции класса.	Статические данные и статические функции класса. Особенности применения статических данных и статических функций. Локальные классы и их применение. Вложенные классы и их применение. Примеры программ с использованием статических, локальных и вложенных классов.
Раздел 7. Использование класса QString для создания класса по работе с большими		

числами. Контейнерные классы. Обработка массивов с использованием класса QVector		
7.1	Тема 1. Пример класса для работы с большими числами.	Пример класса для работы с большими числами. Использование объектов класса QString для представления больших чисел. Реализация операторных функций.
7.2	Тема 2. Краткое описание контейнерных классов.	Краткое описание контейнерных классов. Примеры решения задач с использованием контейнерных классов.
7.3	Тема 3. Пример использования класса QVector для решения задач по обработке массивов.	Пример использования класса QVector для решения задач по обработке массивов.
7.4	Тема 4. Вывод массивов на экран.	Вывод массивов на экран. Организация прокрутки в окне.
Раздел 8. Создание и обработка таблиц. Клиент-серверные приложения (консольные). Клиент-серверные приложения (с графическим интерфейсом)		
8.1	Тема 1. Класс QTableWidgetItem.	Класс QTableWidgetItem. Пример решения задачи по обработке таблицы. Вкладки.
8.2	Тема 2. Основные сведения об организации передачи информации по сети.	Основные сведения об организации передачи информации по сети.
8.3	Тема 3. Класс QSocet.	Класс QSocet. Реализация клиент-серверного консольного приложения.
8.4	Тема 4. Пример организации клиент-серверного приложения с графическим интерфейсом.	Пример организации клиент-серверного приложения с графическим интерфейсом.
Раздел 9. Поток. Построение графиков. Имитация движения на экране		
9.1	Тема 1. Процессы и потоки.	Процессы и потоки. Организация вспомогательных потоков. Синхронизация потоков.
9.2	Тема 2. Построение графиков функций.	Построение графиков функций. Масштабирование. Построение графиков по данным, считанным из файла.
9.3	Тема 3. Класс QTimer.	Класс QTimer.
9.4	Тема 4. Смена изображений на экране по сигналам таймера.	Смена изображений на экране по сигналам таймера.

Лабораторные занятия

№	Темы лабораторных работ
1.	Арифметические и логические выражения. Операторы управления if, if else, switch.
2.	Операторы цикла while, do while, for. Работа с одномерными и многомерными массивами
3.	Функции. Использование арифметических и математических функций при программировании математических задач.
4.	Функции работы со строками. Программирование обработки текстовых данных
5.	Массивы, структуры, объединения. Программирование работы с агрегатными типами данных
6.	Ввод-вывод. Обработка текстовых данных. Программирование в различных системах ввода –вывода.
7.	Ввод-вывод. Обработка бинарных файлов. Программирование в различных системах ввода вывода.
8.	Программирование математических задач, задач численных методов и задач моделирования
9.	Простейшие классы
10.	Конструкторы, деструкторы
11.	Перегрузка функций и операций
12.	Простое и множественное наследование

13.	Виртуальные функции и виртуальные классы
14.	Шаблонные функции и шаблонные классы
15.	Классы динамических структур
16.	Программирование математических задач, задач численных методов и моделирования
17.	Создание классов для работы с большими числами
18.	Примеры программ, использующих контейнерные классы
19.	Решение задач с использованием класса QVector. Организация прокрутки в окне.
20.	Решение задач с использованием класса QTableWidgetItem. Вкладки.
21.	Клиент-серверные приложения (консольные)
22.	Клиент-серверные приложения (с графическим интерфейсом)
23.	Приложения, использующие потоки.
24.	Приложения для построения графиков.
25.	Приложения, имитирующие движение на экране.

4.3. Перечень учебно-методического обеспечения для самостоятельной работы студентов

1. Стили и методы программирования: курс лекций: учебное пособие. Автор: Непейвода Н.Н. Издательство: Интернет-Университет. Информационных Технологий, 2005 г.
2. Введение в программирование: Учебное пособие Авторы: Баженова И.Ю., Сухомлин В.А. Издательство: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007 г.
3. И.В. Виленкин. Высшая математика для студентов экономических, технических, естественно-научных специальностей вузов. Учеб. пособие. Ростов н/Д: Феникс, 2002 г.

При изучении дисциплины используются следующие виды самостоятельной работы:

- ✓ самостоятельный поиск литературы по разделам и темам курса;
- ✓ изучение материала по дополнительным разделам дисциплины;
- ✓ изучение литературы и подготовка к выполнению лабораторных работ;
- ✓ подготовка к экзамену.

5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

5.1. Паспорт фонда оценочных средств по дисциплине

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

Раздел	Темы занятий	Компетенция	Индикаторы освоения	Текущий контроль, неделя
Семестр 1				
Раздел 1	Тема 1. Алгоритм и его свойства.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 1
	Тема 2. Теория структурного программирования.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 1
	Тема 3. Назначение и состав системы программирования.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 2
Раздел 2	Тема 1. Этапы развития языка.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 3
	Тема 2. Стандарты.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 3
	Тема 3. Данные в языке С.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 4
	Тема 4. Операции и их классификация, приоритеты операций.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 4
Раздел 3	Тема 1. Общая характеристика и назначение операторов управления.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 5
	Тема 2. Общее определение и применение функции как самостоятельной программной единицы.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 6
	Тема 3. Массивы: одномерные (векторы), двумерные (матрицы), многомерные массивы. Классы памяти.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 7
	Тема 4. Общие принципы организации систем ввода-вывода.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 8
Рубежный контроль		ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	ИЗ 16
Раздел 4	Тема 1. Важнейшие характеристики ООП.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 9
	Тема 2. Понятие и роль функции в классе. Функции конструкторы и деструкторы их назначение и типы.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 10
	Тема 3. Понятие дружественной функции и дружественного класса. Общее понятие наследования классов.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 11-12
Раздел 5	Тема 1. Общее понятие перегрузки функций. Общие принципы перегрузки операций ввода-вывода в рамках класса.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 13-14

	Тема 2. Ссылки и их роль и назначение при передаче параметров функций.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 15-16
Рубежный контроль		ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	ИЗ 16
Промежуточная аттестация		ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	Экзамен
Семестр 2				
Раздел 6	Тема 1. Полиморфизм в C++. Базовые классы и виртуальные функции – механизм реализации полиморфизма.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 1
	Тема 2. Особенности использования виртуальных функций.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 2
	Тема 3. Шаблоны функций для реализации полиморфизма в C++.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 3
	Тема 4. Статические данные и статические функции класса.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 4
Раздел 7	Тема 1. Пример класса для работы с большими числами.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 5
	Тема 2. Краткое описание контейнерных классов.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 6
	Тема 3. Пример использования класса QVector для решения задач по обработке массивов.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 7
	Тема 4. Вывод массивов на экран.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 8
Рубежный контроль		ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	ИЗ 8
Раздел 8	Тема 1. Класс QTableWidgetItem.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 9
	Тема 2. Основные сведения об организации передачи информации по сети.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 10
	Тема 3. Класс QSocet.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 11
	Тема 4. Пример организации клиент-серверного приложения с графическим интерфейсом.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 12
Раздел 9	Тема 1. Процессы и потоки.	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 13
	Тема 2. Построение графиков функций.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 14
	Тема 3. Класс QTimer.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 15
	Тема 4. Смена изображений на экране по сигналам таймера.		3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	УО 16

Рубежный контроль	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	ИЗ 16
Промежуточная аттестация	ОПК-1 ОПК-3 ОПК-5	3-ОПК-1; У-ОПК-1; В-ОПК-1 3-ОПК-3; У-ОПК-3; В-ОПК-3 3-ОПК-5; У-ОПК-5; В-ОПК-5	Экзамен

5.2. Примерные контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

5.2.1. Оценочные средства для текущего контроля

5.2.1.1. Примерные вопросы для устного опроса

1. Постановка задачи и построение алгоритма.
2. Разработка программы.
3. Трансляция программы.
4. Тестирование программы.
5. Язык С. Лексические структуры языка.
6. Разделители, ограничители, идентификаторы.
7. Константы целого и вещественного типа.
8. Константы символьного и строкового типа.
9. Переменные целого типа.
10. Переменные вещественного типа.
11. Переменные символьного и строкового типа.
12. Приоритет и порядок выполнения операций.
13. Операции отношения, побитовые операции.
14. Операции присваивания, операция SIZEOF.
15. Составной оператор, пустой оператор.
16. Конструкции выбора.
17. Операторы цикла WHILE, DO WHILE.
18. Оператор цикла FOR. Вложенные циклы.
19. Функции. Общие сведения.
20. Прототипы функций.
21. Ввод-вывод в С. Спецификации.
22. Функция PRINTF.
23. Функция SCANF.
24. Функции работы с файлами
25. Ввод-вывод в C++. Общие принципы.

26. Форматирующие функции-элементы.
27. Флаги форматирования.
28. Манипуляторы.
29. Обработка текстовых файлов в C++.
30. Обработка двоичных файлов в C++.
31. Указатели.
32. Массивы. Способы описания массивов.
33. Строки. Функции работы со строками.
34. Структуры.
35. Объединения.
36. Основные понятия объектно-ориентированного программирования в C++.
37. Понятие класса.
38. Режимы доступа.
39. Перегруженные функции.
40. Наследование.
41. Конструкторы.
42. Деструкторы.
43. Конструктор с параметрами.
44. Конструктор копирования.
45. Дружественные функции.
46. Дружественные классы.
47. Аргументы функций, задаваемые по умолчанию.
48. Структуры и классы.
49. Объединения и классы.
50. Наследование классов.
51. Режимы доступа при наследовании.
52. Конструкторы с параметрами при наследовании.
53. Множественное наследование.
54. Перегрузка функций - членов класса.
55. Перегрузка операций.
56. Ключевое слово **this**.
57. Перегрузка операций ввода-вывода. Инсерторы.
58. Перегрузка операций ввода-вывода. Экстракторы.
59. Дружественные функции-операции.
60. Ссылки.
61. Использование ссылочных переменных для перегрузки унарных операций.

62. Виртуальные функции.
63. Использование виртуальных функций.
64. Чистые виртуальные функции.
65. Абстрактные типы.
66. Производные классы и их конструкторы и деструкторы.
67. Конструкторы и деструкторы при множественном наследовании.
68. Виртуальные базовые классы.
69. Операции динамического выделения памяти **new** и **delete**.
70. Виртуальные деструкторы.
71. Шаблоны функций.
72. Шаблоны классов.
73. Статические члены класса.
74. Локальные классы.
75. Вложенные классы.
76. Использование класса QString для работы с большими числами.
77. Пример класса для работы с большими числами.
78. Использование объектов класса QString для представления больших чисел.
79. Реализация операторных функций.
80. Контейнерные классы.
81. Краткое описание контейнерных классов.
82. Примеры решения задач с использованием контейнерных классов.
83. Обработка массивов с использованием класса QVector.
84. Пример использования класса QVector для решения задач по обработке массивов.
85. Вывод массивов на экран.
86. Организация прокрутки в окне.
87. Создание и обработка таблиц.
88. Класс QTableWidgetItem.
89. Пример решения задачи по обработке таблицы.
90. Вкладки.
91. Клиент-серверные приложения (консольные).
92. Основные сведения об организации передачи информации по сети.
93. Класс QSocet.
94. Реализация клиент-серверного консольного приложения.
95. Клиент-серверные приложения (с графическим интерфейсом).
96. Пример организации клиент-серверного приложения с графическим интерфейсом.
97. Потоки.

98. Процессы и потоки.
99. Организация вспомогательных потоков.
100. Синхронизация потоков.
101. Построение графиков.
102. Построение графиков функций.
103. Масштабирование.
104. Построение графиков по данным, считанным из файла.
105. Имитация движения на экране.
106. Класс QTimer.
107. Смена изображений на экране по сигналам таймера.

5.2.2. Оценочные средства для рубежного контроля

5.2.2.1. Примерные темы индивидуального задания

- ✓ Моделирование различных процессов, например физических.
- ✓ Приложение для обработки результатов экспериментов.
- ✓ Программирование численных методов.
- ✓ Решение задач дискретной математики.
- ✓ Создание обучающих программ.
- ✓ Создание программ для контроля знаний по конкретной предметной области.
- ✓ Создание информационных систем или баз данных.
- ✓ Решение экономических задач.
- ✓ Создание клиент-серверных приложений с различными функциями.
- ✓ Создание приложения для работы с интернетом.
- ✓ Решение задач разбора выражений.
- ✓ Создание различных редакторов.
- ✓ Создание программ обработки текстов: анализа, шифровки или сжатия.
- ✓ Игровые программы.

5.2.3. Оценочные средства для промежуточной аттестации

5.2.3.1. Примерные вопросы к экзамену

1. Постановка задачи и построение алгоритма.
2. Разработка программы.
3. Трансляция программы.
4. Тестирование программы.
5. Язык C. Лексические структуры языка.
6. Разделители, ограничители, идентификаторы.

7. Константы целого и вещественного типа.
8. Константы символьного и строкового типа.
9. Переменные целого типа.
10. Переменные вещественного типа.
11. Переменные символьного и строкового типа.
12. Приоритет и порядок выполнения операций.
13. Операции отношения, побитовые операции.
14. Операции присваивания, операция SIZEOF.
15. Составной оператор, пустой оператор.
16. Конструкции выбора.
17. Операторы цикла WHILE, DO WHILE.
18. Оператор цикла FOR. Вложенные циклы.
19. Функции. Общие сведения.
20. Прототипы функций.
21. Ввод-вывод в C. Спецификации.
22. Функция PRINTF.
23. Функция SCANF.
24. Функции работы с файлами
25. Ввод-вывод в C++. Общие принципы.
26. Форматирующие функции-элементы.
27. Флаги форматирования.
28. Манипуляторы.
29. Обработка текстовых файлов в C++.
30. Обработка двоичных файлов в C++.
31. Указатели.
32. Массивы. Способы описания массивов.
33. Строки. Функции работы со строками.
34. Структуры.
35. Объединения.
36. Основные понятия объектно-ориентированного программирования в C++.
37. Понятие класса.
38. Режимы доступа.
39. Перегруженные функции.
40. Наследование.
41. Конструкторы.
42. Деструкторы.

43. Конструктор с параметрами.
44. Конструктор копирования.
45. Дружественные функции.
46. Дружественные классы.
47. Аргументы функций, задаваемые по умолчанию.
48. Структуры и классы.
49. Объединения и классы.
50. Наследование классов.
51. Режимы доступа при наследовании.
52. Конструкторы с параметрами при наследовании.
53. Множественное наследование.
54. Перегрузка функций - членов класса.
55. Перегрузка операций.
56. Ключевое слово **this**.
57. Перегрузка операций ввода-вывода. Инсерторы.
58. Перегрузка операций ввода-вывода. Экстракторы.
59. Дружественные функции-операции.
60. Ссылки.
61. Использование ссылочных переменных для перегрузки унарных операций.
62. Виртуальные функции.
63. Использование виртуальных функций.
64. Чистые виртуальные функции.
65. Абстрактные типы.
66. Производные классы и их конструкторы и деструкторы.
67. Конструкторы и деструкторы при множественном наследовании.
68. Виртуальные базовые классы.
69. Операции динамического выделения памяти **new** и **delete**.
70. Виртуальные деструкторы.
71. Шаблоны функций.
72. Шаблоны классов.
73. Статические члены класса.
74. Локальные классы.
75. Вложенные классы.
76. Использование класса QString для работы с большими числами.
77. Пример класса для работы с большими числами.
78. Использование объектов класса QString для представления больших чисел.

79. Реализация операторных функций.
80. Контейнерные классы.
81. Краткое описание контейнерных классов.
82. Примеры решения задач с использованием контейнерных классов.
83. Обработка массивов с использованием класса QVector.
84. Пример использования класса QVector для решения задач по обработке массивов.
85. Вывод массивов на экран.
86. Организация прокрутки в окне.
87. Создание и обработка таблиц.
88. Класс QTableWidgetItem.
89. Пример решения задачи по обработке таблицы.
90. Вкладки.
91. Клиент-серверные приложения (консольные).
92. Основные сведения об организации передачи информации по сети.
93. Класс QSocet.
94. Реализация клиент-серверного консольного приложения.
95. Клиент-серверные приложения (с графическим интерфейсом).
96. Пример организации клиент-серверного приложения с графическим интерфейсом.
97. Потoki.
98. Процессы и потоки.
99. Организация вспомогательных потоков.
100. Синхронизация потоков.
101. Построение графиков.
102. Построение графиков функций.
103. Масштабирование.
104. Построение графиков по данным, считанным из файла.
105. Имитация движения на экране.
106. Класс QTimer.
107. Смена изображений на экране по сигналам таймера.

5.3. Шкалы оценки образовательных достижений

Рейтинговая оценка знаний является интегральным показателем качества теоретических и практических знаний и навыков студентов по дисциплине и складывается из оценок, полученных в ходе текущего контроля и промежуточной аттестации.

Результаты текущего контроля и промежуточной аттестации подводятся по шкале балльно-рейтинговой системы.

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	Оценка по 4-ех балльной шкале	Оценка ECTS	Требования к уровню освоению учебной дисциплины
90-100	5 – «отлично»	A	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.
85-89	4 – «хорошо»	B	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		C	
70-74		D	
65-69	3 «удовлетворительно»	E	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.
60-64			
Ниже 60	2 «неудовлетворительно»	F	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА

1. Литвиненко Н. А. - Технология программирования на C++ .Год издания: 2010
Издательство: БХВ-Петербург
2. Романов Е.Л. Си++. От дилетанта до профессионала Год издания: 2014 Макс Шлее - Qt4.5
Профессиональное программирование на C++. Год издания:2010
3. Стили и методы программирования: курс лекций: учебное пособие Автор: Непейвода
Н.Н. Издательство: Интернет-Университет Информационных Технологий, 2005 г.
4. Введение в программирование: Учебное пособие Авторы: Баженова И.Ю., Сухомлин В.А.
Издательство: Интернет-Университет Информационных Технологий; БИНОМ.
Лаборатория знаний, 2007 г.
5. Программирование на языке C++ Автор: Павловская Т.А. Издательство: Интернет-
Университет Информационных Технологий, 2010 г.
6. Шлее М. Qt4.5 Профессиональное программирование на C++. СПб.: БХВ-Петербург,
2010.
7. Травова Н.Н. Лабораторный практикум по Qt часть 2(размещен на Server-129 в папке
public/Travova/Qt).
8. Бланшет Ж., Саммерфилд М. Qt4. Программирование GUIна C++. 2-е изд. М.: КУДИЦ-
Пресс, 2007.
9. Земсков Ю. Qt4 на примерах. СПб.: БХВ-Петербург, 2008.
10. Шилдт Г. Полный справочник по C++. 4-е изд. М.: Вильямс, 2006
11. Стефенс Д. Р. C++. Сборник рецептов. М.: КУДИЦ-Пресс, 2007.
12. Саммерфилд М. Qt4. Профессиональное программирование. Разработка кросс-
платформенных приложений на C++. СПб.: Символ-Плюс 2011.
13. Дж. Макконнелл Анализ алгоритмов. Вводный курс. М.: ТЕХНОСФЕРА, 2002.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Н. Вирт. Алгоритмы и структуры данных. СПб.: Невский Диалект, 2001
2. Э. Йодан. Структурное программирование и конструирование программ.М.:Мир,1979
3. Т. Кормен, Ч. Лейзерсон, Р. Ривест. Алгоритмы: Построение и анализ. – СПб, 2003
Д.Э.Кнут. Искусство программирования:Т.1. Основные алгоритмы. – М. Вильямс,
20005.Д.Э.Кнут. Искусство программирования: Т.2. Получисленные алгоритмы. М:
Вильямс, 2000

4. Д.Э. Кнут. Искусство программирования: Т. 3.Сортировка и поиск.– М: Вильямс, 2000
5. Г.С. Уоррен. Алгоритмические трюки для программистов. – СПб, 2003
6. И.В. Виленкин. Высшая математика для студентов экономических, технических, естественно-научных специальностей вузов. Учеб. пособие. Ростов н/Д: Финикс, 2002 г.
7. Дж. Коплиен. Программирование на C++. Классика CS. – СПб.: Питер, 2005
8. Р. Вайнер, Лю Пинсон. C++ изнутри. Пер. с англ. – Киев: «ДиаСофт», 1993
9. М. Эллис, Б. Строуструп. Справочное руководство по языку программирования C++ с комментариями. Пер. с англ. Мю: Мир, 1992
10. А. Мешков, Ю. Тихомиров. Visual C++ и MFC. Программирование для Windows NT и Window 95. Т.1.- СПб.:BHV – Санкт-Петербург, 1997
11. К. Паппас, У. Мюррей. Visual C++. Руководство для профессионалов: пер. с англ. – СПб.:BHV – Санкт-Петербург, 1996

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

1. Intel Pentium 4, 512M RAM, 40G HDD с установленным программным обеспечением: MS WindowsXP, MS Office Pro, Microsoft Visual Studio 6.0
2. IntelPentium 4, 512MRAM, 40GHDD с установленным программным обеспечением: OCLinuxс графической оболочкой GNOM, QtCreator.

LMS И ИНТЕРНЕТ-РЕСУРСЫ:

1. Учебно-методический материал в библиотеке института, ресурсы Интернета, ресурсы электронной библиотеки.
2. Электронный ресурс: intuit.ru (Интернет университет информационных технологий)

7 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Класс ПЭВМ не ниже Intel Pentium 4, 512M RAM, 40G HDD с установленным программным обеспечением: MS WindowsXP, MS Office Pro, Microsoft Visual Studio 6.0 и др.из расчета одна ПЭВМ на одного человека.

Класс ПЭВМ не ниже IntelPentium 4, 512MRAM, 40GHDD с установленным программным обеспечением: OCLinuxс графической оболочкой GNOM, QtCreator.

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В соответствии с требованиями ОС ВО по направлению подготовки 01.03.02 «Прикладная математика и информатика» реализация компетентностного подхода

предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков студентов.

В рамках учебного курса студенты работают с лекциями, рекомендованной литературой, выполняют лабораторные работы, готовятся к экзамену. В процессе подготовки студенты используют программные продукты, инструментальные среды, информационно-справочные системы, информационные источники, размещенные в сети Интернет (официальные сайты, веб-порталы, тематические форумы и телекоммуникации), электронные учебники и учебно-методические пособия.

По дисциплине «Языки и методы программирования» в рабочем учебном плане предусмотрены интерактивные часы для проведения лабораторных занятий. Для реализации интерактивных форм обучения используются учебно-методические материалы (лабораторные практикумы), разработанные сотрудниками кафедры «Прикладной математики».

9. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОРГАНИЗАЦИИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Примерным учебным планом на изучение дисциплины отводится два семестра. В конце семестров предусмотрен экзамен.

В первом семестре изучается стандарт языка C с элементами языка C++, а также изучению основ объектно-ориентированного программирования (ООП) на базе языка C++.

Второй семестр посвящен изучению технологии программирования Qt с применением объектно-ориентированного программирования. В процессе преподавания дисциплины методически целесообразно в каждом разделе курса выделить наиболее важные моменты и акцентировать на них внимание обучающихся.

Рекомендации преподавателю

В первом разделе заострить внимание на понятии алгоритма и его свойствах, разновидностях структур алгоритмов, способы описания алгоритмов, обратить внимание на теоретические аспекты структурного программирования, реализацию основ структурного программирования в языках программирования. Рассмотреть назначение и состав систем программирования. Привести классификацию языков программирования. Дать описание и назначение инструментальной среды Microsoft Visual Studio 6.0. и системы программирования Visual C++ 6.0.

Во втором разделе изложить общие характеристики языка C, его основные элементы и конструкции. Общие принципы построения программы на языке C. Рассмотреть данные, применяемые в языке C, типы констант и переменных, занимаемая ими память и диапазоны

допустимых значений. Уделить внимание программированию вычислений выражений, арифметическим и логическим операциям, а также преобразованию типов в операторах присваивания.

В третьем разделе большое внимание уделить операторам управления, так как они управляют логикой выполнения программы. Следует обратить особое внимание на изложение материала, связанного с использованием операторов цикла. Рассмотреть самостоятельную программную единицу – функцию, способы описания функций, типы функций, механизмы передачи параметров и возврата результата выполнения функции. Также рассмотреть сложные (агрегатные) типы данных: многомерных массивов, структур и объединений. Изложить материал, содержащий рекомендации к применению сложных данных в различных программах. Рассмотреть классы памяти, использование динамической памяти, основные функции работы с динамической памятью. Обратит внимание на использование операторов ввода-вывода при программировании задач, связанных с обработкой данных, расположенных на внешних носителях. Рассмотреть две системы ввода-вывода, операторы и функции ввода вывода, общие принципы обработки текстовой и числовой информации.

В четвертом разделе заострить внимание на понятии ООП, рассмотреть основные понятия - класс и объект, привести важнейшие характеристики ООП. Рассмотреть понятие функции и ее роль в классе. Определить функции конструкторы и деструкторы их назначение и типы. Рассмотреть конструктор по умолчанию, основной конструктор, конструктор копирования. Рассмотреть назначение и использование дружественной функции и дружественного класса. в C++, реализацию доступа к данным класса через дружественные функции. Изложить общее понятие наследования классов. Рассмотреть роль атрибутов и свойств объектов при наследовании, доступ к данным и функциям при наследовании, простое и множественное наследование.

В пятом разделе рассмотреть общие принципы перегрузки функций и операций, особенности перегрузки унарных и бинарных, подчеркнуть эффективность применения данного механизма при написании программ. Уделить внимание правилам и механизму перегрузки операций (операторов) ввода-вывода, их удобство при программировании. Уделить внимание ссылкам, их роли и назначению при передаче параметров функций. Показать использование ссылочных переменных при перегрузке унарных операций и операции индексации. Привести отличие механизма использования ссылок от передачи параметров по значению и через указатели.

В шестом разделе рассмотреть виртуальные функции и виртуальные классы, механизмы их использования при написании эффективных программ, подчеркнуть важность их применения для реализации полиморфизма. Рассмотреть шаблоны функций и шаблоны

классов, их роли для написания программ, обрабатывающих данные различного типа. Рассмотреть статические, локальные и вложенные классы, механизмы их создания и применения.

В седьмом разделе рассказать о реализации класса для работы с большими числами для закрепления навыков объектно-ориентированного программирования. Рассмотреть контейнерные классы и примеры решения задач с использованием этих классов. Разобрать возможности класса `QVector` для решения задач по обработке массивов.

В восьмом разделе рассмотреть решение задач по обработке табличных данных. Рассмотреть организацию передачи данных по локальной сети в консольных приложениях с использованием блокирующих вызовов. Познакомить учащихся с передачей данных с использованием класса `QDataStream`.

В девятом разделе рассмотреть работу многопоточных приложений и синхронизацию потоков. Познакомить учащихся с приемами построения графиков в виде ломаных линий. Рассмотреть класс `QTimer` имитации движения по сигналам таймера

Рекомендации студенту

- ✓ Самостоятельно прорабатывать лекционный материал для более полного усвоения материала;
- ✓ В учебном процессе при выполнении лабораторного практикума эффективно использовать методические пособия и методический материал по темам лабораторных работ;
- ✓ Активно использовать Интернет-ресурсы для получения актуального материала по изучаемой дисциплине;
- ✓ Активно использовать Интернет-ресурсы для обновления инструментальной базы (систем программирования, инструментальных сред и т.д.) при выполнении лабораторных работ.

Программа составлена в соответствии с требованиями ОС ВО НИЯУ МИФИ к обязательному минимуму содержания основной образовательной программы по направлению подготовки 01.03.02 Прикладная математика и информатика.

Программу составил:

Рецензент: