

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)

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ЭЛЕКТРОНИКИ
Кафедра «Вычислительной и информационной техники»

УТВЕРЖДАЮ

Декан ФИТЭ, к.ф-м.н., доцент

_____ **В.С. Холушкин**

«___» _____ **2023 г.**

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ

наименование дисциплины

Направление подготовки (специальность)	09.03.01 Информатика и вычислительная техника
Наименование образовательной программы	Программное и аппаратное обеспечение высокопроизводительных вычислительных систем и сетей
Квалификация (степень) выпускника	бакалавр
Форма обучения	очная

Программа одобрена на заседании кафедры

Зав. кафедрой ВИТ

Протокол № _____ от _____

_____ **В.С. Холушкин**

«___» _____ **2023г.**

г. Саров, 2023г.

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

Заведующий кафедрой ВИТ

В.С. Холушкин

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

Заведующий кафедрой ВИТ

В.С. Холушкин

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

Заведующий кафедрой ВИТ

В.С. Холушкин

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

Заведующий кафедрой ВИТ

В.С. Холушкин

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/
2	32	3	108	16	-	32	60	-	3
ИТОГО	32	3	108	16	-	32	60	-	

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/
3		3	108	16	-	32	24	-	3
ИТОГО		3	108	16	-	32	24	-	36

АННОТАЦИЯ

Курс посвящен изучению теоретических и практические основ алгоритмизации и программирования в объектно-ориентированном подходе разработки программного обеспечения. Изучаются способы и методы разработки эффективных алгоритмов для решения различных прикладных задач. Главная цель преподавания дисциплины – подготовка специалиста, владеющего фундаментальными знаниями и практическими навыками в области основ алгоритмизации и программирования для решения прикладных задач в различных предметных областях.

1. ЦЕЛИ И ЗАДАЧИ ОСТВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В дисциплине «Программирование» 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 C++, Visual C++, Qt. Главная цель преподавания дисциплины – 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

Дисциплина «Программирование» является базовой дисциплиной федерального государственного образовательного стандарта высшего профессионального образования (ОС ВО) по направлениям «Информационные системы и технологии», «Информатика и вычислительная техника» (квалификация – «бакалавр»).

Изучение дисциплины «Программирование» основывается на базе знаний, умений и компетенций, полученных студентами в ходе освоения школьного курса «Информатика» и дисциплины «Алгоритмические языки».

Дисциплина «Программирование» являются базовым теоретическим и практическим основанием всех последующих дисциплин подготовки бакалавра.

3.ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Общепрофессиональные компетенции (ОПК)

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	З-ОПК-8 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения У-ОПК-8 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули В-ОПК-8 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	З-ОПК-2 Знать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, используемых при решении задач профессиональной деятельности У-ОПК-2 Уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности В-ОПК-2 Владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

4.СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

№ п/п	Наименование раздела /темы дисциплины	№ не- дели	Виды учебной работы					
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			16	-	32	60		
Семестр 2								
Раздел 1.								
1.1.	Тема 1. . Основ- ные понятия ООП.	1	2		4	6	УО, За- щита ЛР	5
1.2	Тема 2. Конструк- торы и деструкто- ры	2	2		4	6	Защита ЛР	5
Раздел 2.								

№ п/п	Наименование раздела /темы дисциплины	№ не- дели	Виды учебной работы					
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			16	-	32	60		
2.1	Тема 1. Друже- ственные функ- ции и друже- ственные классы	3,4	2		4	8	Защита ЛР	5
2.2	Тема 2. Наследо- вание классов	5,6	2		4	8	Защита ЛР	4
2.3	Тема 3. Перегруз- ка функций и операций. Пере- грузка операций ввода-вывода	7,8	2		4	8	Защита ЛР	4
2.4	Тема 4. Ссылки. Виртуальные функции и вирту- альные классы	9,10	2		4	8	Защита ЛР	4
Рубежный кон- троль		11	СР					5
Раздел 3.								
3.1	Тема 1.Шаблоны функций и шаб- лоны классов	12,13	2		4	8	Защита ЛР	4
3.2	Тема 2. Статиче- ские, локальные и вложенные клас- сы	14,15	2		4	8	Защита ЛР	4
Рубежный кон- троль		16	СР					5
Промежуточная аттеста- ция			3				-	50
Посещаемость								5
Итого:			16		32	60	-	100

*Сокращение наименований форм текущего, рубежного и промежуточного контроля:

УО – устный опрос

СР – самостоятельная работа(решение задачи на заданную тему)

РГР – расчетно – графическая работа

№ п/п	Наименование раздела /темы дисциплины	№ не- дели	Виды учебной работы					
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			16	-	32	24		
Семестр 3								

№ п/п	Наименование раздела /темы дисциплины	№ не- дели	Виды учебной работы					
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий контроль (форма)*	Максимальный балл (см. п. 5.3)
			16	-	32	24		
Раздел 1.								
1.1.	Тема 1. Класс для работы с боль- шими числами	1	2		4	3	УО, За- щита ЛР	4
1.2	Тема 2. Контей- нерные классы	2	2		4	3	Защита ЛР	4
Раздел 2.								
2.1	Тема 1. Работа с классом QVector	3,4	2		4	3	Защита ЛР	4
2.2	Тема 2. Создание и обработка таб- лиц	5,6	2		4	3	Защита ЛР	4
2.3	Клиент- серверные при- ложения (кон- сольные)	7,8	2		4	3	Защита ЛР	4
2.4	Тема 4. Клиент- серверные при- ложения (с гра- фическим интер- фейсом)	9,10	2		4	3	Защита ЛР	4
Рубежный кон- троль		11	СР					8
Раздел 3.								
3.1	Тема 1. Поток	12,13	2		4	3	Защита ЛР	4
3.2	Тема 2. Построе- ние графиков. Имитация дви- жения на экране	14,15	2		4	3	Защита ЛР	4
Рубежный кон- троль		16	СР					5
Промежуточная аттеста- ция			Э				-	50
Посещаемость								5
Итого:			16		32	24	-	100

*Сокращение наименований форм текущего, рубежного и промежуточного контроля:

УО – устный опрос

СР – самостоятельная работа(решение задачи на заданную тему)

РГР – расчетно – графическая работа

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс (2 семестр)

№	Наименование раздела /темы дисциплины	Содержание
2 семестр		
Раздел 1		
1.1	Тема 1.Основные понятия 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.
1.2	Тема 2.Конструкторы и деструкторы	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.
Раздел 2		
2.1	Тема 1.Дружественные функции и дружественные классы	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.
2.2	Тема 2.Наследование классов	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.
2.3	Тема 3.Перегрузка функций и операций. Перегрузка операций	Общее понятие перегрузки функций. Перегрузка конструкторов. Перегрузка операций в рамках класса. Оператор-функция operator . Механизмы и принципы перегрузки

	ввода-выв	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
2.4	Тема 4. Ссылки. Виртуальные функции и виртуальные классы.	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. Полиморфизм в C++. Базовые классы и виртуальные функции – механизм реализации полиморфизма. Особенности использования виртуальных функций. Чистые виртуальные функции. Виртуальные классы и их применение. Примеры программ с применением виртуальных функций и виртуальных классов.
Раздел 3.		
3.1	Тема 1. Шаблоны функций и шаблоны классов	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.
3.2	Тема 2. Статические, локальные и вложенные классы.	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.

Лекционный курс (3 семестр)

№	Наименование раздела /темы дисциплины	Содержание
3 семестр		
Раздел 1		
1.1	Тема 1.Использование класса QString для создания класса по работе с большими числами	Пример класса для работы с большими числами. Использование объектов класса QString для представления больших чисел. Реализация операторных функций.
1.2	Тема 2.Контейнерные классы.	Краткое описание контейнерных классов. Примеры решения задач с использованием контейнерных классов.
Раздел 2		
2.1	Тема 1.Обработка массивов с использованием класса QVector	Пример использования класса QVector для решения задач по обработке массивов. Вывод массивов на экран. Организация прокрутки в окне.
2.2	Тема 2.Создание и обработка таблиц.	Класс QTableWidgetItem. Пример решения задачи по обработке таблицы. Вкладки.
2.3	Тема 3.Клиент-серверные приложения (консольные)	Основные сведения об организации передачи информации по сети. Класс QSocket. Реализация клиент-серверного консольного приложения.
2.4	Тема 4.Клиент-серверные приложения (с графическим интерфейсом)	Пример организации клиент-серверного приложения с графическим интерфейсом.
Раздел 3		
3.1	Тема 1.Потоки	Процессы и потоки. Организация вспомогательных потоков. Синхронизация потоков.
3.2.	Тема 2.Построение графиков. Имитация движения на экране.	Построение графиков функций. Масштабирование. Построение графиков по данным, считанным из файла. Класс QTimer. Смена изображений на экране по сигналам таймера.

Лабораторные занятия

Лабораторный практикум предусмотрен в каждом из двух семестров. Лабораторный практикум предполагает выполнение лабораторных работ по основным разделам дисци-

плины. Темы лабораторных работ приведены в следующей таблице соответственно по семестрам. Инструментальная среда для выполнения лабораторных работ Microsoft Visual Studio13 и выше.

№	Примерные темы лабораторных занятий (2 семестр)
1	Простейшие классы
2	Конструкторы, деструкторы
3	Перегрузка функций и операций
4	Простое и множественное наследование
5	Виртуальные функции и виртуальные классы
6	Шаблонные функции и шаблонные классы
7	Классы динамических структур
8	Программирование математических задач, задач численных методов и моделирования
№	Примерные темы лабораторных занятий (Третий семестр)
1	Создание классов для работы с большими числами
2	Примеры программ, использующих контейнерные классы
3	Решение задач с использованием класса QVector. Организация прокрутки в окне.
4	Решение задач с использованием класса QTableWidgetItem. Вкладки.
5	Клиент-серверные приложения (консольные)
6	Клиент-серверные приложения (с графическим интерфейсом)
7	Приложения, использующие потоки.
8	Приложения для построение графиков.
9	Приложения, имитирующие движение на экране.

Курсовые работы

По дисциплине «Программирование» предусмотрена курсовая работа. Примерные темы курсовых работ:

- 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.

4.3 Перечень учебно-методического обеспечения для самостоятельной работы студентов

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, курсовых работ;
- подготовка к тестированию, контрольным работам, написанию рефератов;
- подготовка к зачету, экзаменам.

Форма контроля: отчет по лабораторным работам и их защита, защита курсовых работ.

Учебно-методические пособия:

1. Программирование на языке C++ Автор: Павловская Т.А. Издательство: Интернет-Университет Информационных Технологий, 2010 г.
2. Курс лекций по C.C++. Холушкин В.С. Электронный ресурс СарФТИ НИЯУ МИФИ. 2021 г.
3. Основы алгоритмизации 2. Лабораторный практикум. Травова Н.Н. Электронный ресурс СарФТИ НИЯУ МИФИ 2021 г.
4. Лабораторный практикум по Qt часть 2. Травова Н.Н Электронный ресурс СарФТИ НИЯУ МИФИ 2021 г.
5. Курс лекций по Q. Травова Н.Н Электронный ресурс СарФТИ НИЯУ МИФИ 2021 г.

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 следующих таблицах:

Раздел	Темы занятий	Компетенция	Индикаторы освоения	Текущий контроль, неделя
Семестр 2				
Раздел 1	Тема 1. Основные понятия ООП	ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	УО 1
	Тема 2. Конструкторы и деструкторы		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР3
Раздел 2	Тема 1. Дружественные функции и дружественные классы	ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР5
	Тема 2. Наследование классов		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР7
	Тема 3. Перегрузка функций и операций. Перегрузка операций ввода-вывода		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР9
	Тема 4. Ссылки. Виртуальные функции и виртуальные классы		3-ОПК-8; У-ОПК-8; В-ОПК-8 3-ОПК-9; У-ОПК-9; В-ОПК-9	Защита ЛР10
Рубежный контроль		ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	СР 11
Раздел 3	Тема 1. Шаблоны функций и шаблоны классов	ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР13
	Тема 2. Статические, локальные и вложенные классы		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР 15
Рубежный контроль		ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	
Промежуточная аттестация		ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Зачет

Раздел	Темы занятий	Компетенция	Индикаторы освоения	Текущий контроль, неделя
Семестр 3				
Раздел 1	Тема 1. Использование класса QString для создания класса по работе с большими числами	ОПК-2, ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	УО 1
	Тема 2. Контейнерные классы		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР3

Раздел 2	Тема 1. Обработка массивов с использованием класса QVector	ОПК-2,ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР5
	Тема 2 Создание и обработка таблиц		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР7
	Тема 3. Клиент-серверные приложения (консольные)		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР9
	Тема 4. Клиент-серверные приложения (с графическим интерфейсом)		3-ОПК-8; У-ОПК-8; В-ОПК-8 3-ОПК-9; У-ОПК-9; В-ОПК-9	Защита ЛР10
Рубежный контроль		ОПК-2,ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	СР 11
Раздел 3	Тема 1. Поток	ОПК-2,ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР13
	Тема 2. Имитация движения на экране		3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Защита ЛР 15
Рубежный контроль		ОПК-2,ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	
Промежуточная аттестация		ОПК-2,ОПК-8	3-ОПК-2; У-ОПК-2; В-ОПК-2 3-ОПК-8; У-ОПК-8; В-ОПК-8	Экзамен

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (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 (УО)

2 семестр

1. Парадигмы ООП
2. Понятие и назначение класса
3. Конструкторы, деструкторы
4. Наследование
5. Полиморфизм
6. Перегрузка операций и функций
7. Режимы доступа и наследования
8. Виртуальные функции
9. Шаблоны функций и классов
10. Статические, локальные, вложенные классы

3 семестр

1. Класс для работы с большими числами
2. Контейнерные классы
3. Обработка массивов
4. Работа с таблицами

5. Консольные приложения
6. Приложения с графическим интерфейсом
7. Применение и работа с потоками
8. Графика и имитация движения объектов

5.2.1.2. Примерные темы и вопросы для самостоятельной работы (СР)

1. Использование языка программирования C++ для решения прикладных задач из различных предметных областей (математика, физика, экономика и др.)
2. Использование библиотек при программировании на C, C++ для решения задач со сложными типами данных (очереди, стеки, множества, списки и др.)
3. Особенности программирования консольных приложений. Программные средства создания графических интерфейсов.

5.2.2. Оценочные средства для рубежного контроля

5.2.2.1. Примерные задания для решения задач по заданной теме

3.1. Разработать класс **String** для работы со *строками*. Класс должен содержать *конструктор по умолчанию, основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса, а также операторную *функцию присваивания*. Кроме этого перегрузить операторную функцию - (*унарный минус*), которая *упорядочивает строку по убыванию*.

3.2. Создать класс **Circle** для работы с *плоскими окружностями*. В качестве члена данного задается длина радиуса окружности. Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса, а также операторную *функцию присваивания*. Дополнительно перегрузить для элементов этого класса операторные функции для двух операций \sim (*побитовое отрицание*) и $!$ (*логическое отрицание*), возвращающие *длину окружности* и *площадь круга*, соответственно.

3.3. Разработать класс **Rectangle** для работы с *плоскими прямоугольниками*, стороны которого параллельны осям координат. В качестве членов-данных задать координаты двух точек прямоугольника (левую верхнюю точку и нижнюю правую). Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса. Дополнительно перегрузить операторную функцию для операции $!$ (*логическое отрицание*),

которая возвращает *площадь вписанного в исходный прямоугольник серединного четырехугольника* (четырёхугольника, получаемого в результате последовательного соединения середин сторон исходного прямоугольника).

3.4. Разработать класс **ThreeAngle** для работы с *плоскими треугольниками*. В качестве членов-данных задаются координаты трех вершин треугольника. Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса. Дополнительно перегрузить *логическую* операторную функцию для операции *|* (*побитовая дизъюнкция*), которая определяет *принадлежность точки с заданными координатами треугольнику*.

3.5. Разработать класс **ThreeAngle** для работы с *плоскими треугольниками*. В качестве членов-данных задаются длины трех сторон треугольника. Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса, а также операторную *функцию присваивания*. Дополнительно перегрузить операторную функцию для операции *!* (*логическое отрицание*), которая возвращает *длины всех высот треугольника*.

3.6. Разработать класс **ThreeAngle** для работы с *плоскими треугольниками*. В качестве членов-данных задаются длины трех сторон треугольника. Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса, а также операторную *функцию присваивания*. Дополнительно перегрузить операторную функцию для операции *!* (*логическое отрицание*), которая возвращает *длины всех биссектрис треугольника*.

3.7. Разработать класс **ThreeAngle** для работы с *плоскими треугольниками*. В качестве членов-данных задаются длины трех сторон треугольника. Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса, а также операторную *функцию присваивания*. Дополнительно перегрузить операторную функцию для операции *!* (*логическое отрицание*), которая возвращает *длины всех медиан треугольника*.

3.8. Разработать класс **Line** для работы с объектами типа *отрезок*. Члены-данные этого класса определяют координаты концов отрезка на плоскости. Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса. Дополнительно перегрузить операторную функцию для операции **!**, *определяющую длину отрезка*.

3.9. Разработать класс **Line** для работы с объектами типа *отрезок*. Члены-данные этого класса определяют координаты концов отрезка на плоскости. Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса. Дополнительно перегрузить *логическую* операторную функцию для операции **|** (*побитовая дизъюнкция*), *определяющую параллельность двух отрезков*.

3.10. Разработать класс **Line** для работы с объектами типа *отрезок*. Члены-данные этого класса определяют координаты концов отрезка на плоскости. Класс должен содержать *основной конструктор и конструктор копирования*. Перегрузить в этом классе операторные функции, которые обеспечивают *ввод/вывод* элементов класса. Дополнительно перегрузить *логическую* операторную функцию для операции **|** (*побитовая дизъюнкция*), *определяющую перпендикулярность двух отрезков*.

1. Написать программу с окном типа QMainWindow, которое считывает текст из файла и выводит его на экран, выравнивая длину каждой строки до заданной. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Длину строки считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.
2. Написать программу с окном типа QMainWindow, которое считывает текст из файла, выводит его на экран и подсчитывает, сколько раз заданный символ содержится в тексте. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Символ считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.

3. Написать программу с окном типа QMainWindow, которое считывает текст из файла, выводит его на экран и подсчитывает, сколько раз заданное слово содержится в тексте. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Слово считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.
4. Написать программу с окном типа QMainWindow, которое считывает текст из файла и выводит его на экран, затем выводит на экран первые *n* символов этого файла. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Количество символов считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.
5. Написать программу с окном типа QMainWindow, которое считывает текст из файла и выводит его на экран, затем выводит на экран первые *n* слов этого файла. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Количество слов считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.
6. Написать программу с окном типа QMainWindow, которое считывает текст из файла и выводит его на экран, затем выводит на экран первые *n* строк этого файла. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Количество строк считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.
7. Написать программу с окном типа QMainWindow, которое считывает текст из файла и выводит его на экран, затем выводит на экран первые *n* символов каждой строки. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Количество символов считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.
8. Написать программу с окном типа QMainWindow, которое считывает текст из файла и выводит его на экран, затем выводит на экран те строки, которые начинаются с заданной последовательности символов. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Символы считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.

9. Написать программу с окном типа QMainWindow, которое считывает текст из файла и выводит его на экран, затем выводит на экран первые n слов каждой строки. Если в строке меньше слов, то выводится вся строка. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Количество слов считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.
10. Написать программу с окном типа QMainWindow, которое считывает текст из файла и выводит его на экран, затем выводит на экран те строки, которые заканчиваются заданной последовательностью символов. Имя файла должно считываться с помощью специального диалогового окна для открытия файлов. Символы считывать с помощью вспомогательного диалогового окна, которое выводится на экран при нажатии на кнопку на панели инструментов.

5.2.3. Оценочные средства для промежуточной аттестации

5.2.3.2. Примерные вопросы к зачету и экзамену

Вопросы к зачету по курсу “Программирование” второй семестр.

1. Основные понятия объектно-ориентированного программирования в C++.
2. Понятие класса.
3. Режимы доступа.
4. Перегруженные функции.
5. Наследование.
6. Конструкторы.
7. Деструкторы.
8. Конструктор с параметрами.
9. Конструктор копирования.
10. Дружественные функции.
11. Дружественные классы.
12. Аргументы функций, задаваемые по умолчанию.
13. Структуры и классы.
14. Объединения и классы.
15. Наследование классов.
16. Режимы доступа при наследовании.
17. Конструкторы с параметрами при наследовании.
18. Множественное наследование.
19. Перегрузка функций - членов класса.

20. Перегрузка операций.
21. Ключевое слово **this**.
22. Перегрузка операций ввода-вывода. Инсерторы.
23. Перегрузка операций ввода-вывода. Экстракторы.
24. Дружественные функции-операции.
25. Ссылки.
26. Использование ссылочных переменных для перегрузки унарных операций.
27. Виртуальные функции.
28. Использование виртуальных функций.
29. Чистые виртуальные функции.
30. Абстрактные типы.
31. Производные классы и их конструкторы и деструкторы.
32. Конструкторы и деструкторы при множественном наследовании.
33. Виртуальные базовые классы.
34. Операции динамического выделения памяти **new** и **delete**.
35. Виртуальные деструкторы.
36. Шаблоны функций.
37. Шаблоны классов.
38. Статические члены класса.
39. Локальные классы.
40. Вложенные классы.

Вопросы к экзамену по курсу “Программирование” третий семестр.

1. Использование класса QString для работы с большими числами.
2. Пример класса для работы с большими числами.
3. Использование объектов класса QString для представления больших чисел.
4. Реализация операторных функций.
5. Контейнерные классы.
6. Краткое описание контейнерных классов.
7. Примеры решения задач с использованием контейнерных классов.
8. Обработка массивов с использованием класса QVector.
9. Пример использования класса QVector для решения задач по обработке массивов.
10. Вывод массивов на экран.
11. Организация прокрутки в окне.
12. Создание и обработка таблиц.
13. Класс QTableWidgetItem.

14. Пример решения задачи по обработке таблицы.
15. Вкладки.
16. Клиент-серверные приложения (консольные).
17. Основные сведения об организации передачи информации по сети.
18. Класс QSocet.
19. Реализация клиент-серверного консольного приложения.
20. Клиент-серверные приложения (с графическим интерфейсом).
21. Пример организации клиент-серверного приложения с графическим интерфейсом.
22. Потоки.
23. Процессы и потоки.
24. Организация вспомогательных потоков.
25. Синхронизация потоков.
26. Построение графиков.
27. Построение графиков функций.
28. Масштабирование.
29. Построение графиков по данным, считанным из файла.
30. Имитация движения на экране.
31. Класс QTimer.
32. 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 освоения 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 изла-

			гает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.
85-89	4 – «хорошо»	В	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		С	
70-74		Д	
65-69	3 – «удовлетворительно»	Е	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»	Ф	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 ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА:

1. Литвиненко Н. А. - Технология программирования на С++ .Год издания: 2010 Издательство: БХВ-Петербург
2. Романов Е.Л. Си++. От дилетанта до профессионала Год издания: 2014
3. Макс Шлее - Qt4.5 Профессиональное программирование на С++. Год издания: 2010
4. Стили и методы программирования: курс лекций: учебное пособие
Автор: Непейвода Н.Н.Издательство: Интернет-Университет Информационных Технологий, 2005 г.
5. Введение в программирование: Учебное пособие Авторы: Баженова И.Ю., Сухомлин В.А. Издательство: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007 г.

6. Программирование на языке C++ Автор: Павловская Т.А. Издательство: Интернет-Университет Информационных Технологий, 2010 г.
7. Шлее М. Qt4.5 Профессиональное программирование на C++. СПб.: БХВ-Петербург, 2010.
8. Травова Н.Н. Лабораторный практикум по Qt часть 2(размещен на Server-129 в папке public/Travova/Qt).
9. Бланшет Ж., Саммерфилд М. Qt4. Программирование GUI на C++. 2-е изд. М.: КУДИЦ-Пресс, 2007.
10. Земсков Ю. Qt4 на примерах. СПб.: БХВ-Петербург, 2008.
11. Шилдт Г. Полный справочник по C++. 4-е изд. М.: Вильямс, 2006
12. Стефенс Д. Р. C++. Сборник рецептов. М.: КУДИЦ-Пресс, 2007.
13. Саммерфилд М. Qt4. Профессиональное программирование. Разработка кросс-платформенных приложений на C++. СПб.: Символ-Плюс 2011
14. 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
4. Д.Э.Кнут. Искусство программирования: Т.1. Основные алгоритмы. – М.: Вильямс, 2000
5. Д.Э.Кнут. Искусство программирования: Т.2. Получисленные алгоритмы. М.: Вильямс, 2000
6. Д.Э. Кнут. Искусство программирования: Т. 3. Сортировка и поиск. – М.: Вильямс, 2000
7. Г.С. Уоррен. Алгоритмические трюки для программистов. – СПб, 2003
8. И.В. Виленкин. Высшая математика для студентов экономических, технических, естественно-научных специальностей вузов. Учеб. пособие. Ростов н/Д: Финикс, 2002 г.
9. Дж. Коппиен. Программирование на C++. Классика CS. – СПб.: Питер, 2005
10. Р. Вайнер, Лю Пинсон. C++ изнутри. Пер. с англ. – Киев: «ДиаСофт», 1993
11. М.Эллис, Б.Строуструп. Справочное руководство по языку программирования C++ с комментариями. Пер. с англ. Мю: Мир, 1992
12. А. Мешков, Ю. Тихомиров. Visual C++ и MFC. Программирование для Windows NT и Window 95. Т.1.- СПб.: BHV – Санкт-Петербург, 1997
13. К. Паппас, У. Мюррей. Visual C++. Руководство для профессионалов: пер. с англ. – СПб.: BHV – Санкт-Петербург, 1996

Учебно-методический материал в библиотеке института, ресурсы Интернета, ресурсы электронной библиотеки.

Электронный ресурс: intuit.ru (Интернет университет информационных технологий)

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Класс ПЭВМ не ниже Intel Pentium 4, 512M RAM, 40G HDD с установленным программным обеспечением: MS WindowsXP, MS Office Pro, Microsoft Visual Studio 613 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 09.03.01 «Информатика и вычислительная техника» 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 зачету. 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.

9. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОРГАНИЗАЦИИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Предлагается

- 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 09.03.01 Информатика и вычислительная техника.

Автор(ы) _____ В.С.Холушкин

Рецензенты _____ Н.В.Фролова

Согласовано:

Зав. кафедрой ВИТ _____ В.С.Холушкин

Руководитель ОП _____ В.С.Холушкин