

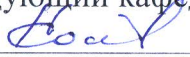
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Радиотехнические системы»

«СОГЛАСОВАНО»

Заведующий кафедрой РТС

 / В.И. Кошелев /
17 мая 2023 г.

«УТВЕРЖДАЮ»

Проректор по учебной работе

 / А.В. Корячко /
20 мая 2023 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ПРОГРАММИРОВАНИЕ НА C++

Направление подготовки
11.03.01 Радиотехника

Направленность (профиль) подготовки
Аппаратно-программная инженерия радиолокационных и навигационных систем

Квалификация выпускника — бакалавр

Форма обучения — очная

Рязань 2023

Аннотация

Курс «Программирование на C++» посвящен созданию масштабируемых приложений на C++ и более углубленно раскрывает основные концепции объектно-ориентированного программирования. В течении курса будут разобраны основные паттерны проектирования и целесообразность их использования. В заключении, у каждого студента будет функционирующее приложение с возможностью масштабирования и добавления нового функционала, без изменения уже созданного кода.

1. Цели и задачи

Цель дисциплины:

Закрепление и расширение навыков программирования на языке программирования C++, изучение паттернов проектирования масштабируемых приложений.

Задачи дисциплины:

- углублённое изучение языка программирования C++;
- знакомство с паттернами проектирования. применение этих паттернов для написания приложений;
- использовать систему контроля версий для систематизации и хранения кода.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенций
ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-4.1 Использует при решении задач профессиональной деятельности современные информационные технологии и программные средства, в том числе отечественного производства
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.1 Понимает требования к алгоритмам, суть процесса алгоритмизации задач
	ОПК-5.2 Выполняет разработку алгоритмического и программного обеспечения для решения прикладных задач

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины, обучающиеся должны:

- знать:
 - синтаксис и особенности языка C++;
 - принципы объектно-ориентированного программирования;
 - основы SQL;

- наиболее распространенные паттерны проектирования;
- уметь:
 - программировать на C++;
 - применять паттерны программирования для разработки приложений;
 - отлаживать код;
- владеть:
 - языком программирования C++;
 - системой контроля версий Git;
 - средой коллективной разработки Jira.

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

№	Тема (раздел)	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семи- нары	Лабора- торные работы	Самост. работа
1	Системы контроля версий и коллективной разработки	1	1	1	1
2	Синтаксис языка программирования C++	1	1	1	1
3	Концепции объектно-ориентированного программирования	2	2	2	1
4	UML-диаграммы	2	2	2	2
5	SOLID	2	2	2	2
6	Умные указатели	2	2	2	2
7	Контейнеры и функции стандартных библиотек	2	2	2	2
8	Паттерны проектирования	2	2	2	2
9	Юнит-тесты	2	2	2	2
Итого часов		16	16	16	15
Подготовка к экзамену		0 час.			
Общая трудоемкость		72 час., 2 зач. ед.			

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

Семестр 5 (осенний)

1. Системы контроля версий и коллективной разработки
Изучение систем Git и Jira для обеспечения командной работы, отслеживания версий и создания задач и контроля выполнения.

2. Синтаксис языка программирования C++
Повторения синтаксиса языка программирования C++: типы данных, наиболее распространенные команды и другие синтаксические конструкции.

3. Концепции объектно-ориентированного программирования
Применение концепций объектно-ориентированного программирования для масштабирования приложения и построения иерархии кода для целей последующей модернизации.

4. UML-диаграммы
Применение UML-диаграмм для визуализации архитектуры приложения, обсуждения и изложения архитектуры приложения и компонентов приложения при командной разработке.

5. SOLID
Принцип единственной ответственности. Принцип открытости/закрытости. Принцип подстановки Лискова. Принцип разделения интерфейса. Принцип инверсии зависимостей.

6. Умные указатели
Концепция умных указателей. Проблема циклических ссылок. Примеры реализации.

7. Контейнеры и функции стандартных библиотек
Основные типы контейнеров: последовательные; ассоциативные; адаптеры.

8. Паттерны проектирования
Концепция паттерна проектирования. Наиболее распространенные ПП: фабрика, адаптер, наблюдатель, декоратор.

9. Юнит-тесты
Концепция юнит-тестов. Примеры реализации покрытия кода тестами для проверки корректности их работы.

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

Компьютерный класс с доской, проектором или телевизором, подключенный к сети.

6. Перечень рекомендуемой литературы

Основная литература

1. Язык программирования C++ [Текст] = The C++ Programming Language, [учеб. пособие для вузов] /Бьерн Страуструп ; пер. с англ. под ред. Н. Н. Мартынова. -М., БИНОМ, 2017

Дополнительная литература

1. Параллельное программирование многопоточных систем с разделяемой памятью [Текст] : учеб. пособие для вузов / А. Г. Тормасов .— М : Физматкнига, 2014 .— 208 с.

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

1. <http://cs.mipt.ru>
2. <http://acm.mipt.ru>

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

Для контроля и коррекции знаний, обучающиеся могут использовать компьютерное тестирование, в том числе на сайте www.judge.mipt.ru. В процессе самостоятельной работы обучающихся возможно использование любые среды программирования.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий курс информатики, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Показателем владения материалом служит умение решать задачи. Для формирования умения применять теоретические знания на практике студенту необходимо решать как можно больше задач. При решении задач каждое действие необходимо аргументировать, ссылаясь на известные теоретические сведения.

При подготовке к сдаче зачета необходимо повторять ранее изученные основные понятия. Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к лектору или преподавателю. Обязательным требованием является выполнение домашних работ, которые оформляются в специально отведённой для этого тетради и систематически сдаются на проверку.

Промежуточный контроль знаний проводится в виде контрольных работ, на которых студенту предлагается решить несколько задач, а также студенту в ходе освоения курса необходимо выполнить две домашние индивидуальные работы с их последующей защитой.

Оценочные материалы по дисциплине (модулю)

по направлению: Радиотехника

профиль подготовки: Аппаратно-программная инженерия радиолокационных и навигационных систем
Сетевое обучение
Кафедра «Алгоритмов и технологий программирования»

курс: 3

квалификация: бакалавр

Семестр, форма промежуточной аттестации: 5 (осенний) – зачет

Разработчик: Г.Д. Свечников, преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины (модуля)

Код и наименование компетенции	Индикаторы достижения компетенций
ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-4.1 Использует при решении задач профессиональной деятельности современные информационные технологии и программные средства, в том числе отечественного производства
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.1 Понимает требования к алгоритмам, суть процесса алгоритмизации задач
	ОПК-5.2 Выполняет разработку алгоритмического и программного обеспечения для решения прикладных задач

2. Показатели оценивания компетенция

В результате освоения дисциплины Практикум по программированию на C++, обучающиеся должны:

- знать:
 - синтаксис и особенности языка C++;
 - принципы объектно-ориентированного программирования;
 - основы SQL;
 - наиболее распространенные паттерны проектирования;
- уметь:
 - программировать на C++;
 - применять паттерны программирования для разработки приложений;
 - отлаживать код;
- владеть:

- языком программирования C++;
- системой контроля версий Git;
- средой коллективной разработки Jira.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. Устройство компилятора. Понятие компиляции, виды оптимизирующей компиляции.
2. Внутреннее представление программы и его построение. Управляющий граф, построение и использование.
3. Особенности архитектур с точки зрения оптимизаций.
4. Архитектуры современных микропроцессоров.
5. Машинная модель.
6. Методика оптимизаций. Выбор оптимальной платформы. Профилирование и определение наиболее затратных фрагментов кода. Выделение фрагментов кода, способных дать максимальный прирост производительности. Выбор оптимального алгоритма.
7. Динамический и статический профилировщики программ.
8. Общие стратегии оптимизаций. Загрузка программы.
9. Общие стратегии оптимизаций Подсистема ввода-вывода.
10. Общие стратегии оптимизаций Переключение контекста.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Общие стратегии оптимизаций Системные вызовы.
2. Общие стратегии оптимизаций Обращение к БД.
3. Общие стратегии оптимизаций Динамическое связывание.
4. Общие стратегии оптимизаций Обращения к памяти.
5. Общие стратегии оптимизаций Обращения к сети.
6. Общие стратегии оптимизаций Цепочки зависимостей.
7. Проектирование с учетом оптимизации. Использование STL.
8. Проектирование с учетом оптимизации. Метапрограммирование.
9. Проектирование с учетом оптимизации. Указатели vs ссылки. Конструкторы, деструкторы. Возвращаемые значения. Поля и методы. Наследование.
10. Проектирование с учетом оптимизации. Потоки. Потоки ввода-вывода. Исключения и другие способы обработки ошибок. RTTI. Раскручивание стека. Move-семантика.
11. Оптимизация на уровне написания кода. Массивы структур vs структуры массивов. Векторные операции.
12. Оптимизация на уровне написания кода. Intrinsic. Inlining. Кэширование. Учет кэширования в процессоре. Выравнивание данных. Операции с плавающей точкой. Целочисленные операции. Ассемблерные вставки. Проверка диапазонов. Побитовые операции.
13. Межпроцедурный анализ программ.

Билет 1

Межпроцедурные оптимизации.

Билет 2

Мелко-зернистые оптимизации.

Билет 3

Сбор общих подвыражений, удаление избыточных операций чтения из памяти.

Критерии оценивания

«зачтено» – выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты;

«не зачтено» – выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины, а также собственными конспектами занятий по предмету. Зачет проводится по итогам текущей активности в ходе занятий, защиты инициативной курсовой работы, и путем организации специального опроса, проводимого в простой устной форме, в виде беседы преподавателя и студента.