

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы
радиотехники и компьютерных
технологий**

Е.А. Белянко

	Рабочая программа дисциплины (модуля)
по дисциплине:	Основы разработки приложений с использованием C++
по направлению:	Прикладные математика и физика
профиль подготовки:	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра информационных систем
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 30 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: А.В. Лотоцкий, канд. техн. наук, старший преподаватель

Программа обсуждена на заседании кафедры информационных систем 29.04.2021

Аннотация

В курсе рассматриваются основные подходы к разработке прикладного программного обеспечения с использованием языка программирования C++. Дается обзор основных языковых возможностей C++, рассматриваются вопросы разработки классов и наследования, управления памятью и ресурсами, оптимизации и производительности, а также основных алгоритмах и классах стандартной библиотеки STL. Дается общее представление о методологиях разработки сложных вычислительных систем. Обсуждаются различные типы архитектур программного обеспечения. Изучаются основы СУБД и языка SQL.

Большое внимание уделяется практической работе, как основному методу усвоения информации по данному направлению. На практических занятиях рассматриваются примеры задач и выполняются самостоятельные задания.

Для успешного прохождения курса необходимо посещение и конспектирование изучаемого материала, выполнение практических заданий и самостоятельная работа с дополнительными литературными источниками.

1. Цели и задачи

Цель дисциплины

- освоение студентами базовых знаний и принципов построения современных платформенно-независимых приложений и систем, разрабатываемых на языке программирования C++ с использованием реляционных баз данных, современных программно-аппаратных архитектур, а также получение полного обзора возможностей основных средств разработки.

Задачи дисциплины

- изучение основ программирования на C++;
- изучение основных алгоритмов STL;
- обучение принципам оценивания сложности алгоритмов;
- обучение студентов основным принципам проектирования приложений и систем;
- обзор принципов разработки с использованием СУБД;
- обучение студентов принципам оптимизации вычислений.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научной, технической и (или) иной информации	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности

обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основы программирования на C++;
- условия и основы применения СУБД в программном обеспечении;
- основные способы и средства разработки программного обеспечения.

уметь:

- применять C++ и SQL на практике;
- разрабатывать оптимальную архитектуру приложения;
- оценивать и эффективно использовать программно-аппаратную архитектуру вычислителя для решения практических задач.

владеть:

- основами программирования на C++;
- принципами разработки программного обеспечения;
- основам реляционных баз данных и языка SQL;
- методами оценки алгоритмической сложности;
- методами оптимизации вычислительных задач.

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) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Основы программирования на C++		7	7	16
2	Базовые алгоритмы STL, сложность алгоритмов		2	2	11
3	Принципы разработки программного обеспечения		5	5	11
4	Основы реляционных СУБД, основы SQL		7	7	16
5	Примеры решения задач		9	9	21
Итого часов			30	30	75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 7 (Осенний)

1. Основы программирования на C++

Вводный курс программирования на языке C++. Дается общее представление о назначении языка, способах организации ввода и вывода данных, типах переменных, операторах и функциях, структурах и классах, методах и шаблонах, организации работы с памятью и ресурсами.

Наследование и виртуальные методы. Модификаторы доступа к членам класса.

2. Базовые алгоритмы STL, сложность алгоритмов

Обзор основных классов, функций и алгоритмов STL. Массивы, векторы, контейнеры, итераторы. Алгоритмы поиска и сортировки.

Понятие вычислительной сложности алгоритмов, использование O-нотации. Оценка сложности алгоритмов по времени выполнения и по используемой памяти на примерах STL-алгоритмов сортировки, поиска по массивам и контейнерам.

Оптимизация использования памяти и ресурсов.

3. Принципы разработки программного обеспечения

Вводный курс в теорию построения программного обеспечения. В курсе рассматриваются: частотный принцип, принцип модульности, идеи функциональной избыточности и избирательности, данные и поведение «по умолчанию».

Основы архитектуры программного обеспечения: выделение компонентов программного обеспечения, выбор структурных элементов и их интерфейсов, их укрупнение и объединение. Шаблоны проектирования: Delegation, Interface, Builder, Lazy initialization, Singleton, Adapter, Observer. Архитектурные шаблоны: клиент-серверная архитектура, «толстый» и «тонкий» клиент, трехзвенная архитектура.

Основы совместной разработки: системы контроля версий, комментирование и документирование программного кода. Модульное и интеграционное тестирование.

4. Основы реляционных СУБД, основы SQL

Основные функции СУБД, компоненты и классификация СУБД. ACID-требования к транзакционным системам.

Таблицы и представления данных. Последовательности. Язык запросов SQL. Транзакции. Хранимые процедуры.

5. Примеры решения задач

Разбор и анализ практических задач на примерах:

- сферической тригонометрии;
- перевода координат между различными системами;
- определения количества движущихся баллистических объектов.

Практическое применение СУБД на примере разработки структуры базы данных в PostgreSQL для поддержания каталога космических объектов.

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

Стандартная аудитория, оснащенная мультимедийным оборудованием (проектор, звуковая система).

6. Перечень рекомендуемой литературы

Основная литература

1. Bjarne Stroustrup. Programming principles and practice using C++. Addison Wesley. 2014. 1312 pg. ISBN-13: 978-0321992789.
2. Яцек Галовиц. C++17 STL Стандартная библиотека шаблонов. — СПб.: Питер, 2018. — 432 с.

Дополнительная литература

1. Кент Бек. Экстремальное программирование. Разработка через тестирование. Питер. 2020. 224с. ISBN: 978-5-4461-1439-9
2. Численные методы [Текст] : учеб. пособие для вузов / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. — 2-е изд. — М. ; СПб. : Физматлит, 2001, 2002. — 632 с.
3. С. Дасгупта, Х. Пападимитриу, У. Вазирани. Алгоритмы. МЦНМО. 2014. 320с. ISBN: 978-5-4439-0236-4
4. Ван Вик Кристофер Дж., Седжвик Р. Алгоритмы на C++. Вильямс. 2019. 1056с. ISBN: 978-5-907144-21-7
5. Бьюли Алан. Изучаем SQL. Символ-Плюс. 2016. 312с. ISBN: 978-5-93286-051-9, 5-93286-051-0, 0-596-00727-2

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

1. <http://lib.mipt.ru> – электронная библиотека физтеха
2. <http://www.edu.ru> – федеральный портал «российское образование»
3. <https://en.cppreference.com/w/> – описание принятых и обсуждаемых стандартов C и C++, описание библиотек классов и их методов, функций, практик применения
4. <https://code-live.ru/tag/cpp-manual/> – самоучитель основам программирования на C++

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

Мультимедийные технологии, презентации PowerPoint.

В процессе практических занятий используются IDE-среды для разработки (Visual Studio, QtCreator), СУБД PostgreSQL, библиотека шаблонов STL, фреймворк Qt, система коллективной разработки Mercurial.

Примеры расчетов типовых задач также приводятся в среде QtCreator и Visual Studio.

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

Успешное освоение курса требует большой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой.

Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы,
- проработку учебного материала (по конспектам семинаров, учебной и научной литературе);
- решение задач, предлагаемых студентам на практических занятиях,
- выполнение практических работ в выбранных программных средах,
- подготовку к контрольным, самостоятельным работам и тестам.

Руководство и контроль за самостоятельной работой студента осуществляется в результате анализа итогов контрольных, самостоятельных работ, а также индивидуальных консультаций.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра информационных систем
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

Разработчик: А.В. Лотоцкий, канд. техн. наук, старший преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

2. Показатели оценивания компетенций

В результате изучения дисциплины «Основы разработки приложений с использованием C++» обучающийся должен:

знать:

- основы программирования на C++;
- условия и основы применения СУБД в программном обеспечении;
- основные способы и средства разработки программного обеспечения.

уметь:

- применять C++ и SQL на практике;
- разрабатывать оптимальную архитектуру приложения;
- оценивать и эффективно использовать программно-аппаратную архитектуру вычислителя для решения практических задач.

владеть:

- основами программирования на C++;
- принципами разработки программного обеспечения;
- основам реляционных баз данных и языка SQL;
- методами оценки алгоритмической сложности;
- методами оптимизации вычислительных задач.

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

Темы практических заданий:

1. Основы программирования на C++:

- a. создание проекта в Visual Studio и первой программы «Hello, World!»;
 - b. объявление переменных, чтение стандартного потока ввода, запись в стандартный поток вывода с использованием cin/cout и Xscanf/Xprintf;
 - c. создание иерархии классов (Animal->Mammal->Cat/Dog), реализация виртуальных методов (eat/drink/sleep/run).
2. Методы оценки алгоритмической сложности, базовые алгоритмы:
- a. реализация алгоритма пузырьковой сортировки;
 - b. оценка сложности полученного алгоритма.
3. Принципы разработки программного обеспечения:
- a. разделение программного кода из пп. 1 и 2 на независимые файлы, содержащие библиотеки классов и процедуры;
 - b. модификация MSVS-проекта для работы с библиотеками;
 - c. реализация шаблонов Builder, Lazy initialization, Singleton.
4. Знакомство с SQL на примере PostgreSQL:
- a. консольное и интерактивное подключение к СУБД;
 - b. создание таблиц для хранения объектов из п.1 (create table ...);
 - c. заполнение и изменение данных (insert/update ...);
 - d. удаление данных и таблиц (delete/truncate/drop);
 - e. создание простейших SQL скриптов.
5. Знакомство с Qt;
- a. создание проекта в QtCreator;
 - b. перенос кода из пп. 1-3 в qt-проект;
 - c. знакомство с signal/slot подходом Qt;
 - d. знакомство в QSqlDatabase, подключение к БД из п. 4;
 - e. чтение и запись данных из БД средствами Qt/
6. Знакомство с Mercurial:
- a. клонирование существующего репозитория;
 - b. модификация файлов, commit, push;
 - c. извлечение измененных данных, разрешение конфликтов;
 - d. работа в ветках, создание и слияние.
- За каждое задание студент получает оценку в соответствии с таблицей «Критерии оценивания».

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

Для сдачи дифференцированного зачета студенты выполняют практическое задание, решение которого предполагает применение как минимум следующих разделов курса:

1. Целые и действительные числа, операторы ввода и вывода;
2. Условные операторы;
3. Циклы for/foreach, while, do;
4. Функции;
5. Структуры и классы, наследование, виртуальные методы;
6. Массивы, контейнеры, словари и множества;
7. Алгоритмы STL

Критерии оценивания

Оценка отлично (10) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8) выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7) 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.

Оценка хорошо (6) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5) выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2) выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1) 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

Собственно задание формулируется каждому студенту за неделю до проведения промежуточной аттестации в виде задания на программирование. Текст решения принимается в виде исходного кода, выполненного на языке C++, для проекта под Visual Studio. Предоставляемое решение должно компилироваться без ошибок, результаты исполнения должны соответствовать поставленной задаче.

Дифференцированный зачет выставляется по результатам защиты практических заданий в соответствии с таблицей критерии оценивания.