

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

УТВЕРЖДЕНО

**директор института -
заместитель директора ФАКТ
М.А. Кудров**

	Рабочая программа дисциплины (модуля)
по дисциплине:	Продвинутое программирование на C++
по направлению:	Прикладные математика и физика
профиль подготовки:	Беспилотные авиационные системы Физтех-школа авиационных и цифровых технологий кафедра технологий проектирования сложных технических систем
курс:	2
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: В.С. Бугаев, преподаватель

Программа обсуждена на заседании кафедры технологий проектирования сложных технических систем
05.03.2024

Аннотация

Курс "Продвинутое программирование на C++" посвящен созданию масштабируемых приложений на C++ , углубленно раскрывает основные концепции ООП. В течении курса будут разобраны основные шаблоны проектирования приложения на языке C++ с учетом современных стандартов языка. Практическая часть заключается в командном выполнении проекта, связанного с созданием приложения по тематике ПИШ с возможностью масштабирования и добавления нового функционала.

1. Цели и задачи

Цель дисциплины

- закрепление и расширение навыков программирования на ЯП C++, изучение паттернов проектирования масштабируемых приложений.

Задачи дисциплины

- повторение и углублённое изучение ЯП 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 компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
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-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
	ОПК-4.3 Умеет составлять аннотации, рефераты, библиографические перечни и обзоры информации в области своей профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- синтаксис и особенности языка C++, принципы ООП, наиболее распространенные шаблоны проектирования на языке C++.

уметь:

- программировать на C++, использовать шаблоны проектирования 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) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Отличие C от C++. Компиляция программ. Инкапсуляция. Манглирование имен. Lvalue	4	2		2
2	Семантика перемещения. RAII. Правило 0/5. SmartPointers	3	3		3
3	Перегрузка операторов, наследование и полиморфизм. Pure virtual calls. Pimpl. RTTI	2	2		2
4	Исключения. Гарантии безопасности исключений. Линия Калаба	2	2		2
5	Проектирование. SOLID	2	3		3
6	Шаблонная специализация. Инстанцирование. Вывод типов и свертка ссылок. SFINAE	2	2		2
7	Итераторы. Типы итераторов	2	2		2
8	Последовательные контейнеры и контейнерные адаптеры	2	2		2
9	Ассоциативные контейнеры	2	2		2
10	Алгоритмы и функторы	2	2		2
11	Метапрограммирование	2	3		3
12	Системы контроля версий и коллективной разработки	2	2		2
13	Фреймворк QtCreator	3	3		3
Итого часов		30	30		30
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

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

Семестр: 4 (Весенний)

1. Отличие C от C++. Компиляция программ. Инкапсуляция. Манглирование имен. Lvalue

Язык C и его проблемы. C++ объединение данных и методов. Обобщение типов и функций. Указатели и ссылки. Процесс компиляции программ на C++. Инкапсуляция. Нарушения инкапсуляции. New и delete. Провисание ссылок. Decaying и lvalues. Cdecl и алиасы using. Манглирование. Правила перегрузки. Пространства имён. Агрегатная инициализация. Списки инициализации и делегация. Деструкторы. Копирование и присваивание. RVO и explicit.

2. Семантика перемещения. RAII. Правило 0/5. SmartPointers

Владение ресурсом. RAII. Указатель-подобность. Семантика перемещения и правые ссылки. Перемещающие конструкторы и операторы. Правило пяти и правило нуля. Умные указатели.

3. Перегрузка операторов, наследование и полиморфизм. Pure virtual calls. Pimpl. RTTI

Статические функции и друзья. Приведение типов. Функции и производительность. Pimpl и делтеры. Консистентность. Сравнения и spaceship. Экзотические операторы. Наследование. Полиморфизм. Pure virtual calls. Статическое и динамическое связывание. Перегрузка виртуальных функций. Закрытое наследование. Ромбовидные схемы и виртуальное наследование. RTTI и dynamic_cast.

4. Исключения. Гарантии безопасности исключений. Линия Калаба

Стандартная иерархия исключений. Нейтральность. Ложное чувство безопасности. Гарантии безопасности. Линия Калба. Noexcept. Присваивание Степанова.

5. Проектирование. SOLID

Проектирование. UML. SOLID. Единственная ответственность. Связность. Принцип открытости и закрытости. Инцидентные структуры данных. Подстановка Лисков. Ковариантность к генерализации. Разделение интерфейса. Инверсия зависимостей.

6. Шаблонная специализация. Инстанцирование. Вывод типов и свертка ссылок. SFINAE

Идея шаблонной специализации. Соглашение о namespace detail. Идея характеристик типов. Инстанцирование и специализации. Ленивость инстанцирования. Частичная специализация. Auto и decltype. Немного о сигнатурах функций. Идиомы for-auto и AAA. Свёртка ссылок. Неуниверсальные ссылки. Снова for-auto и AAARR. Прозрачная оболочка и decltype(auto). Совершенный проброс и std::forward. SFINAE. Интегральные константы и определители типов.

7. Итераторы. Типы итераторов

begin, end и ADL. Превосходство итераторов.

Категории итераторов. Константные и обратные итераторы. Невалидные итераторы

Правила инвалидации.

8. Последовательные контейнеры и контейнерные адаптеры

Деки. Списки. Адаптеры.

Ортогональность адаптеров. Маски. Строки. string_view и span.

9. Ассоциативные контейнеры

Ассоциативные контейнеры. Ментальная модель хеша. Оптимизации хеш-таблиц. Открытая адресация. Квадратные скобки. Множества и группы. Мультимножества. Упорядоченные множества.

10. Алгоритмы и функторы

Захват в теле класс. std::function и стирание типов. Алгоритмы и суффиксы. Копирование и transform. Параллельные версии.

11. Метапрограммирование

Перестановки. Алгоритмы над перестановками. Простые метапрограммы. Квадранты вычислений. Константность и литералы. Указатели и массивы на этапе компиляции. Constexpr control flow. Начало обсуждения constexpr функций. Ограничения на constexpr функции. Идиома throw. Consteval и constinit.

12. Системы контроля версий и коллективной разработки

Jira. Confluence. Git и основные операции.

13. Фреймворк QtCreator

Особенности фреймворка. Системы автоматизации сборки cmake/qmake. Модули Qt. Построение простейших графических приложений с использованием Qt.

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)

На лекциях используются мультимедийные технологии, включая демонстрацию презентаций.

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 "Продвинутое программирование на C++", 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 дифференцированного 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:	Прикладные математика и физика
профиль подготовки:	Беспилотные авиационные системы Физтех-школа авиационных и цифровых технологий кафедра технологий проектирования сложных технических систем
курс:	<u>2</u>
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Разработчик: В.С. Бугаев, преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности
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-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
	ОПК-4.3 Умеет составлять аннотации, рефераты, библиографические перечни и обзоры информации в области своей профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации

2. Показатели оценивания компетенций

В результате изучения дисциплины «Продвинутое программирование на C++» обучающийся должен:

знать:

- синтаксис и особенности языка C++, принципы ООП, наиболее распространенные шаблоны проектирования на языке C++.

уметь:

- программировать на C++, использовать шаблоны проектирования 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. Процессы препроцессинга, компиляции и линковки.
2. Применение идиомы RAII в умных указателях.
3. Применение идиомы SFINAE.
4. Типы итераторов в C++. Недостатки итераторов.
5. Вывод типов в шаблонах.
6. Правила разрешения перегрузок.
7. Сравнение последовательных контейнеров. Алгоритмическая сложность основных операций.
8. Сравнение ассоциативных контейнеров. Алгоритмическая сложность основных операций.
9. Принципы ООП. SOLID.
10. lvalue и gvalue ссылки. Правило пяти.

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

1. Типы итераторов. Инвалидация итераторов.
2. Особенности фреймворка Qt. Отличия систем автоматизации сборки cmake/qmake.
3. Виды ассоциативных и последовательных контейнеров. Различия между ними.
4. Идиома RAII и примеры использования в C++
5. Идиома SFINAE и примеры использования в C++
6. Идиома PIMPL и примеры использования в C++
7. Принципы SOLID и их применение в C++
8. Rvalue -ссылки. Правило пяти и правило нуля. Идеальная передача.
9. Манглирование имен и ADL.
10. Исключения и другие способы обработки ошибок. RTTI. Раскручивание стека.
11. Процесс сборки программы в исполняемый файл. Препроцессинг, компиляция и линковка.
12. Метапрограммирование и constexpr-функции. Примеры метапрограмм.
13. Виды умных указателей. Отличия unique_ptr от shared_ptr и weak_ptr.

Билет 1

1. Идиома RAII и примеры использования в C++.
2. Виды умных указателей. Отличия unique_ptr от shared_ptr и weak_ptr.

Билет 2

1. Метапрограммирование и constexpr-функции. Примеры метапрограмм.
2. Идиома SFINAE и примеры использования в C++

Билет 3

1. Манглирование имен и ADL.
2. Процесс сборки программы в исполняемый файл. Препроцессинг, компиляция и линковка.

Критерии оценивания

Оценка «отлично (10)» выставляется обучающемуся, если показавшему всесторонние, систематизированные, глубокие знания предмета и в ходе беседы он верно и детально ответил на четыре (4) произвольных вопроса из выше приведенного перечня. Детальный ответ предполагает верные ответы на все уточняющие вопросы. Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «отлично (9)» выставляется обучающемуся, если в ходе беседы он верно, но не исчерпывающее детально ответил на четыре (4) произвольных вопроса из выше приведенного перечня (мог не ответить на некоторые уточняющие вопросы). Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «отлично (8)» выставляется обучающемуся, если в ходе беседы он верно, но не исчерпывающее детально ответил на четыре (4) произвольных вопроса из выше приведенного перечня (не ответил на уточняющие вопросы).

Оценка «хорошо (7)» выставляется обучающемуся, если в ходе беседы он верно и достаточно детально ответил на три (3) произвольных вопроса из выше приведенного перечня. Детальный ответ предполагает верные ответы на все уточняющие вопросы. Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «хорошо (6)» выставляется обучающемуся, если в ходе беседы он верно, но не исчерпывающее детально ответил на три (3) произвольных вопроса из выше приведенного перечня (не ответил на некоторые уточняющие вопросы). Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «хорошо (5)» выставляется обучающемуся, если в ходе беседы он верно и достаточно детально ответил на два (2) произвольных вопроса из выше приведенного перечня. Детальный ответ предполагает верные ответы на все уточняющие вопросы.

Оценка «удовлетворительно (4)» выставляется обучающемуся, если в ходе беседы он верно и достаточно детально ответил на один (1) произвольный вопрос из выше приведенного перечня. Детальный ответ предполагает верные ответы на все уточняющие вопросы. Подготовка и защита инициативной курсовой работы является преимуществом.

Оценка «удовлетворительно (3)» выставляется обучающемуся, если в ходе беседы он верно, но не исчерпывающее детально ответил на один (1) произвольный вопрос из выше приведенного перечня (не ответил на уточняющие вопросы).

Оценка «неудовлетворительно (2)» выставляется обучающемуся, если в ходе беседы он не смог ответить ни на один произвольный вопрос из выше приведенного перечня, но смог ответить на наводящие вопросы и вопросы с «подсказками».

Оценка «неудовлетворительно (1)» выставляется обучающемуся, если в ходе беседы он не смог ответить ни на один произвольный вопрос из выше приведенного перечня, а так же ни на один наводящий вопрос.

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

Во время проведения дифференциального зачета обучающиеся могут пользоваться программой дисциплины, а также собственными конспектами занятий по предмету.

Дифференцированный 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.