

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

УТВЕРЖДЕНО

**Директор высшей школы
программной инженерии
А.В. Малеев**

	Рабочая программа дисциплины (модуля)
по дисциплине:	Паттерны проектирования
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 5 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 48 час.

Всего часов: 108, всего зач. ед.: 3

Программу составил: А.В. Созыкин, канд. техн. наук, преподаватель

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 28.06.2023

Аннотация

Курс "Паттерны проектирования" посвящен созданию масштабируемых приложений на C++ и более углубленно раскрывает основные концепции ООП. 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 ЯП C++, изучение паттернов проектирования масштабируемых приложений.

Задачи дисциплины

- 1) Повторение и углублённое изучение ЯП C++.
- 2) Знакомство с паттернами проектирования. Применение этих паттернов для написания приложений.
- 3) 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
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
	УК-2.2 Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1 Владеет на практике методологией составления научно-технических отчетов (проектов)
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ОПК-7.1 Обладает навыками создания и выполнения тестовых сценариев для выявления ошибок в программном обеспечении
	ОПК-7.3 Умеет выявлять узкие места в процессе разработки и предлагать методы и инструменты для его оптимизации
ПК-2 Способен формализовать и алгоритмизировать поставленную задачу	ПК-2.1 Способен формализовать и алгоритмизировать поставленную задачу
	ПК-2.2 Владеет методами и приемами формализации и алгоритмизации задач
ПК-3 Способен проектировать,	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания

разрабатывать, интегрировать, проверять на работоспособность программное обеспечение

ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- синтаксис и особенности языка C++, принципы ООП, 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) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Системы контроля версий и коллективной разработки	3	2		5
2	Синтаксис ЯП C++ (повторение)	3	3		5
3	Концепции объектно-ориентированного программирования (ООП)	3	2		5
4	UML-диаграммы	3	3		5
5	SOLID	4	4		5
6	Умные указатели	3	3		5
7	Контейнеры и функции стандартных библиотек	4	3		6
8	Паттерны проектирования	4	6		6
9	Юнит-тесты	3	4		6
Итого часов		30	30		48
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 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 ЯП C++ (повторение)

Повторения синтаксиса ЯП C++: типы данных, наиболее распространенные команды и другие синтаксические конструкции.

3. Концепции объектно-ориентированного программирования (ООП)

Применение концепций ООП 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. Паттерны проектирования, [учеб. пособие для вузов] / Эр. Фримен, Эл. Фримен . — Санкт-Петербург, Питер, 2017.— URL: <https://ibooks.ru/bookshelf/354827/reading> (дата обращения: 26.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

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 "Паттерны проектирования", 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:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии
курс:	3
квалификация:	бакалавр
Семестр, формы промежуточной аттестации: 5 (осенний) - Дифференцированный зачет	
Разработчик:	А.В. Созыкин, канд. техн. наук, 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-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
	УК-2.2 Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1 Владеет на практике методологией составления научно-технических отчетов (проектов)
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ОПК-7.1 Обладает навыками создания и выполнения тестовых сценариев для выявления ошибок в программном обеспечении
	ОПК-7.3 Умеет выявлять узкие места в процессе разработки и предлагать методы и инструменты для его оптимизации
ПК-2 Способен формализовать и алгоритмизировать поставленную задачу	ПК-2.1 Способен формализовать и алгоритмизировать поставленную задачу
	ПК-2.2 Владеет методами и приемами формализации и алгоритмизации задач
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования

2. Показатели оценивания компетенций

В результате изучения дисциплины «Паттерны проектирования» обучающийся должен:

знать:

- синтаксис и особенности языка C++, принципы ООП, 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. Поток. Поток 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.

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.