

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:

Проректор по учебной работе,
качеству образования – первый
проректор

_____ Хагуров Т.А.

подпись

« 29 » августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
Б1. О.23 Объектно-ориентированное программирование и шаблоны
проектирования

Направление подготовки 01.03.02 Прикладная математика и информатика

Профиль Современные методы машинного обучения и компьютерного зрения

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Объектно-ориентированное программирование и шаблоны проектирования» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.03.02 Прикладная информатика.

Программу составил(и):

А.С.Жук, доцент КВТ

И.О. Фамилия, должность, ученая степень, ученое звание



подпись

подпись

Рабочая программа обсуждена на заседании кафедры вычислительных технологий протокол № 1 от «26» 08 2025г.

Заведующий кафедрой (разработчика)

Т. А. Приходько



Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 01 от «28» августа 2025 г.

Председатель УМК факультета

А. В. Коваленко



Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»,
e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетного образовательного учреждения высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Цель дисциплины – изучение объектно-ориентированной парадигмы программирования и отработка навыков ОО проектирования.

1.2 Задачи дисциплины

- Изучение объектно-ориентированной парадигмы программирования.
- Изучение языка программирования Java
- Получение практического опыта ОО проектирования приложения.
- Получение практического опыта ОО проектирования предметной области
- Изучение ООП паттернов проектирования и архитектурных паттернов.
- Научиться строить описание архитектуры приложения с помощью диаграммы классов и диаграммы последовательностей
- Отработка умений реализации заданного архитектурного паттерна.
- Отработка навыков построения web-сервиса средствами JVM языков на примере фреймворков Spring, SpringBoot.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование и шаблоны проектирования» относится к дисциплинам обязательной части.

Дисциплина в значительной степени **взаимодействует для формирования компетенций** с дисциплинами:

- Бэкенд-разработка;
- Кроссплатформные десктоп приложения;

Дисциплина является базовой для дисциплин: микросервисная архитектура, системы искусственного интеллекта, разработка мобильных приложений.

Требованием к «входным» знаниям является знания основных алгоритмов и структур данных, понимание основ веб-разработки, разработки десктопных приложений.

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: **Data Engineer (Инженер по данным)**

Задачи:

- Проектирование и построение ETL-процессов
- Создание и оптимизация хранилищ данных
- Обеспечение качества и доступности данных
- Настройка инфраструктуры для обработки больших данных
- Интеграция разрозненных источников данных
- Работа с данными в области природопользования, медицины, связи и телекоммуникаций

Роль 2: **ML Engineer (Инженер МО)**

Задачи:

- Реализация ML-моделей в продуктивных системах
- Оптимизация производительности и масштабирование моделей
- Разработка ML-пайплайнов и автоматизация процессов
- Мониторинг качества моделей в продуктиве
- Интеграция ML-решений с бизнес-приложениями

Роль 3: MLOps (Специалист по эксплуатации ИИ)

Задачи:

- Автоматизация процессов обучения и развертывания моделей
- Мониторинг производительности ML-систем
- Управление версиями моделей и данных
- Обеспечение CI/CD для ML-проектов
- Оптимизация вычислительных ресурсов

1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций: ОПК-2.1; ОПК-2.2; ОПК-5.1, PL-2.1

ОПК-2	<i>Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач</i>
ОПК-2.1	Способен применять системный подход к анализу предметной (проблемной) области, выявлению требований к реализации алгоритмов решения прикладных задач Знает стандарт UML2.5 для описания архитектуры веб-сервиса и ООП архитектуры предметной области Знает структуру технического задания Умеет реализовать сервис по требованиям в команде Умеет применять методы рефакторинга и обнаруживать антипаттерны в коде
ОПК-2.2	Применяет современный математический аппарат при построении моделей в различных областях человеческой деятельности Владеет способностью составить модель предметной области
ОПК-5	<i>Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</i>
ОПК-5.1	Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности Знает паттерны проектирования и архитектурные паттерны Умеет применять методы рефакторинга и обнаруживать антипаттерны в коде Владеет способностью разрабатывать веб сервисы на Java Spring, SpringBoot Владеет способностью реализовать MVC приложения с нуля без фреймворков
PL-2	<i>(II) Способен применять JVM-совместимые языки программирования для решения задач в области ИИ</i>
PL-2.1	Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений Применяет основные библиотеки для решения рутинных задач в серверном программировании: ввод-вывод, применение простейших примитивов многопоточного программирования, интеграция с базами данных. Понимает модель памяти Java и способен поддерживать приложения с высоким параллелизмом и конкуренцией. Понимает алгоритмы сборки мусора и способен оптимизировать сборку мусора. Имеет представление о том, как построить взаимодействие со специализированными хранилищами данных эффективно, через инструменты разработчика для JVM-совместимых языков

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач. ед. (72 часа), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)				
		5				
Контактная работа, в том числе:	52,2	52,2				
Аудиторные занятия (всего):	50	50				
Занятия лекционного типа	16	16				
Лабораторные занятия	34	34				
Занятия семинарского типа (семинары, практические занятия)						
Иная контактная работа:	2,2	2,2				
Контроль самостоятельной работы (КСР)	2	2				
Промежуточная аттестация (ИКР)	0,2	0,2				
Самостоятельная работа, в том числе:	19,8	19,8				
Курсовая работа						
Выполнение индивидуальных заданий	16	16				
Реферат						
Подготовка к текущему контролю	3,8	3,8				
Контроль:						
Общая трудоемкость	час.	72	72			
	в том числе контактная работа	52,2	52,2			
	зач. ед	2	2			

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 5 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1.	Введение в разработку Java	16	4		8	4
2.	MVC приложение	16	4		8	4
3.	Рефакторинг и паттерны проектирования	16	4		8	4
4.	Архитектурные паттерны	18	4		10	4
ИТОГО по разделам дисциплины		66	16		34	16
	Контроль самостоятельной работы (КСР)	2				
	Промежуточная аттестация (ИКР)	0,2				
	Подготовка к текущему контролю	3,8				
Общая трудоемкость по дисциплине		72				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Введение в разработку Java	Технология Java, ее кроссплатформенность. Язык Java. Структура программы. ООП.	ЛР
2.	Введение в разработку Java	ООП. Отношения между классами. Диаграмма классов. Коллекции.	ЛР
3.	MVC приложение	Паттерны проектирования. Классификация. Паттерны стратегия и наблюдатель.	ЛР
4.	MVC приложение	Диаграмма последовательностей. MVC приложение.	ЛР
5.	Рефакторинг и паттерны проектирования	Антипаттерны и методы рефакторинга.	ЛР
6.	Рефакторинг и паттерны проектирования	Паттерны проектирования	ЛР
7.	Архитектурные паттерны	MVC. MVP. MVVM. Принципы Solid.	ЛР
8.	Архитектурные паттерны	Spring. SpringBoot.	ЛР

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Введение в разработку Java	Введение в язык. Запуск приложения. ООП.	ЛР
2.	Введение в разработку Java	Коллекции.	ЛР
3.	Введение в разработку Java	Наследование. Рефакторинг. Классы данных. Классы коллекций сущностей предметной области. Диаграмма классов.	ЛР
4.	Введение в разработку Java	Защита лаб	
5.	MVC приложение	Классы модели. Работы с хранилищами данных.	ЛР
6.	MVC приложение	Построение GUI	ЛР
7.	MVC приложение	Построение MVC приложения CRUD обработки одной сущности	ЛР
8.	MVC приложение	Защита лаб	ЛР
9.	Рефакторинг и паттерны проектирования	Рефакторинг классов работы с хранилищами данных. Паттерны стратегия, декоратор, одиночка, адаптер.	ЛР
10.	Рефакторинг и паттерны проектирования	Построение общей ООП модели работы с двумя сущностями. Реализация MVC CRUD обработки второй сущности.	ЛР
11.	Рефакторинг и паттерны проектирования	Построение ООП модели предметной области из трёх сущностей. Реализация MVC приложения	ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
12.	Рефакторинг и паттерны проектирования	Защита лаб	ЛР
13.	Архитектурные паттерны	Типы MVC архитектур. Реализация и отличия.	ЛР
14.	Архитектурные паттерны	Архитектура MVP.	ЛР
15.	Архитектурные паттерны	Spring, springBoot.	ЛР
16.	Архитектурные паттерны	Командная разработка зачётного проекта.	ЛР
17.	Архитектурные паттерны	Защита зачётного проекта.	ЛР

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

Курсовая работа не предусмотрена.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №7 от 07.05.2025
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №7 от 07.05.2025

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательных технологий: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

– Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

– Информационно-коммуникационные технологии (ИКТ) – расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

– Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

– Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

– Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

– Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

– Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

– проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «название дисциплины».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме оценки лабораторных работ к проекта к **зачету**.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Введение в разработку Java	ОПК-5.1, PL-2.1	ЛР 1-3	Проект с кейсом от индустриального партнера
2	MVC приложение	ОПК-3.1, PL-2.1	ЛР 4-6	Проект с кейсом от индустриального партнера
3	Рефакторинг и паттерны проектирования	ОПК-2.1; ОПК-2.2; ОПК-5.1, PL-2.1	ЛР 7-9	Проект с кейсом от индустриального партнера
4	Архитектурные паттерны	ОПК-2.1; ОПК-2.2; ОПК-5.1, PL-2.1	ЛР 10-12	Проект с кейсом от индустриального партнера

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **зачтено**):

ОПК-2	Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач
ОПК-2.1	Способен применять системный подход к анализу предметной (проблемной) области, выявлению требований к реализации алгоритмов решения прикладных задач Знает стандарт UML2.5 для описания архитектуры веб-сервиса и ООП архитектуры предметной области Знает структуру технического задания Умеет реализовать сервис по требованиям в команде

	Умеет применять методы рефакторинга и обнаруживать антипаттерны в коде
ОПК-2.2	Применяет современный математический аппарат при построении моделей в различных областях человеческой деятельности Владеет способностью составить модель предметной области
ОПК-5	<i>Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</i>
ОПК-5.1	Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности Знает паттерны проектирования и архитектурные паттерны Умеет применять методы рефакторинга и обнаруживать антипаттерны в коде Владеет способностью разрабатывать веб сервисы на Java Spring, SpringBoot Владеет способностью реализовать MVC приложения с нуля без фреймворков
PL-2	<i>(II) Способен применять JVM-совместимые языки программирования для решения задач в области ИИ</i>
PL-2.1	Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений Применяет основные библиотеки для решения рутинных задач в серверном программировании: ввод-вывод, применение простейших примитивов многопоточного программирования, интеграция с базами данных. Понимает модель памяти Java и способен поддерживать приложения с высоким параллелизмом и конкуренцией. Понимает алгоритмы сборки мусора и способен оптимизировать сборку мусора. Имеет представление о том, как построить взаимодействие со специализированными хранилищами данных эффективно, через инструменты разработчика для JVM-совместимых языков

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Зачетно-экзаменационные материалы для промежуточной аттестации (зачёт)

1. Распишите основные принципы ООП. Приведите пример работающего кода для иллюстрации выполнения или невыполнения этих принципов.
2. Какие отношения между классами могут быть? Опишите, что они из себя представляют в предметной области, в коде и на диаграмме классов. Покажите пример кода с реализацией этих отношений.
3. Расскажите, что такое ассоциация между классами, приведите частные случаи, покажите примеры. Расскажите, какие паттерны проектирования используют данное отношение. Покажите примеры работающего кода с реализацией данного отношения.
4. Расскажите, как Вы понимаете принцип - предпочитайте делегацию наследованию. Покажите примеры кода, покажите, в каком случае имеет смысл использовать любое из отношений. Расскажите, какие паттерны иллюстрируют данный подход.

5. Расскажите, какие Вам известны способы создания экземпляров классов, покажите примеры некачественного и качественного подхода(работающий код). Расскажите, какие паттерны проектирования, описывающие эти подходы, Вам известны.
6. Расскажите ООП структуру Вашего языка программирования. Расскажите, какие методы принято переопределять и в каком случае.
7. Какие паттерны проектирования Вам известны, что это такое, как их принято классифицировать? Раскройте взаимосвязь паттернов проектирования, архитектурных паттернов и фреймворков. Покажите на примерах.
8. Расскажите, как составлять диаграмму классов, что на ней принято отмечать и каким образом. Раскройте все составляющие, покажите взаимосвязь диаграммы классов и кода.
9. Какие структурные паттерны проектирования Вам известны? Перечислите и расскажите базовые принципы всех паттернов. Раскройте подробно два паттерна, покажите проблему, покажите предлагаемое решение на диаграммах и в коде. Укажите связь этих паттернов с другими известными Вам паттернами.
10. Какие порождающие паттерны проектирования Вам известны? Перечислите и расскажите базовые принципы всех паттернов. Раскройте подробно два паттерна, покажите проблему, покажите предлагаемое решение на диаграммах и в коде. Укажите связь этих паттернов с другими известными Вам паттернами.
11. Какие поведенческие паттерны проектирования Вам известны? Перечислите и расскажите базовые принципы всех паттернов. Раскройте подробно два поведенческих паттерна, покажите проблему, покажите предлагаемое решение на диаграммах и в коде. Укажите связь этих паттернов с другими известными Вам паттернами.
12. Расскажите про подходы в сравнении объектов на равенство и на больше-меньше в руби. Расскажите, какие принципы ООП и паттерны проектирования используются в реализации данных подходов.
13. Расскажите, как иллюстрируется взаимодействие объектов на диаграмме последовательности. Покажите тривиальный пример, далее покажите пример диаграммы последовательностей, иллюстрирующий взаимодействие объектов в рамках любого архитектурного паттерна.
14. Архитектурный паттерн MVC. Расскажите базовый подход, расскажите, какие паттерны проектирования используются при его реализации. Покажите взаимодействие объектов в рамках реализации некоторых операций CRUD на диаграмме последовательностей. Покажите, в каких фреймворках он применяется.
15. Архитектурный паттерн MVP. Расскажите проблематику в двух подходах в MVC, которая привела к появлению данного архитектурного паттерна. Расскажите базовый подход. Покажите взаимодействие объектов в рамках реализации некоторых операций CRUD на диаграмме последовательностей. Покажите, в каких фреймворках он применяется.
16. Архитектурный паттерн MVVM. Расскажите проблематику, которая привела к появлению данного архитектурного паттерна. Расскажите базовый подход. Покажите

взаимодействие объектов в рамках реализации некоторых операций CRUD на диаграмме последовательностей. Покажите, в каких фреймворках он применяется.

17. Архитектурные паттерны. Расскажите о всех архитектурных паттернах, которые Вам известны, базовые подходы, в каких языках и фреймворках применяются. Раскройте паттерн, применённый в фреймворке Вашей задачи. Покажите взаимодействие объектов в рамках реализации некоторых операций CRUD на диаграмме последовательностей.

18. Что такое рефакторинг? Когда его требуется применять? Приведите примеры методов рефакторинга и примеры антипаттернов, которые данные методы позволяют исправить.

19. Раскройте антипаттерны: Неуместная близость, завистливые функции, классы данных, одержимость элементарными типами, ленивый класс. Раскройте почему они считаются антипаттернами, приведите примеры кода, покажите методы рефакторинга, позволяющие их исправить. Укажите паттерны разработки, позволяющие не допускать таких ошибок.

20. Раскройте антипаттерны: дублирование кода, длинный метод, длинный класс, длинный список параметров, повторы условных операторов, альтернативные классы с разными интерфейсами. Раскройте почему они считаются антипаттернами, приведите примеры кода, покажите методы рефакторинга, позволяющие их исправить. Укажите паттерны разработки, позволяющие не допускать таких ошибок.

21. Раскройте антипаттерны: стрела дробью, расходящиеся модификации, посредник, теоретическая общность, параллельные иерархии наследования. Покажите, как они взаимосвязаны. Раскройте почему они считаются антипаттернами, приведите примеры кода, покажите методы рефакторинга, позволяющие их исправить. Укажите паттерны разработки, позволяющие не допускать таких ошибок, или приводящие к ним.

22. Раскройте принцип единой ответственности. Покажите примеры антипаттернов его нарушающих, паттернов, его реализующих. Покажите на работающих примерах.

23. Раскройте принцип открытости - закрытости. Покажите примеры антипаттернов его нарушающих, паттернов, его реализующих. Покажите на работающих примерах.

24. Раскройте принцип подстановки Барбары Лисков. Покажите примеры антипаттернов его нарушающих, паттернов, его реализующих. Покажите на работающих примерах.

25. Раскройте принцип подстановки разделения интерфейсов. Покажите примеры антипаттернов его нарушающих, паттернов, его реализующих. Покажите на работающих примерах.

26. Раскройте принцип инверсии зависимостей. Покажите примеры антипаттернов его нарушающих, паттернов, его реализующих. Покажите на работающих примерах.

27. Принципы SOLID. Зачем нужны, о чём они, как взаимосвязаны с паттернами и антипаттернами? Покажите на примерах. Как реализуются в Вашей задаче и в выбранном Вами фреймворке.

Задача – подготавливается заранее. Выбрать тему – предметную область для автоматизации, построить ER модель предметной области - 3-4 таблицы в 3НФ. На JavaSpring построить web-приложение, реализующее операции CRUD для всех выбранных сущностей.

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации, определяющие процедуры оценивания на экзамене:

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является зачет. Студенты обязаны сдать зачет в соответствии с расписанием и учебным планом.

ФОС промежуточной аттестации состоит из заданий и результатов текущего контроля.

Форма проведения зачета: письменно.

Преподавателю предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи зачета заносится преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и проекта.

Критерии оценки:

Зачтено – выполнено 60% лабораторных работ (на оценку зачтено) и выполнен проект.

Не зачтено – выполнено менее 60% лабораторных работ или не выполнен проект.

Методические рекомендации, определяющие процедуры оценивания лабораторных работ:

Процедура оценивания лабораторных работ проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

По каждой лабораторной работе оформляется отчет. Отчеты сдаются на проверку руководителю в течение курса по мере их выполнения, и защищаются студентами в установленном порядке.

При защите отчета студенту могут быть заданы вопросы и дополнительные задания по сути лабораторной работы, в том числе из списка контрольных вопросов к данной лабораторной работе. При неудовлетворительной оценке знаний студента по теме данного отчета, студент возвращается к повторному изучению соответствующих материалов, после чего допускается к повторной защите. Неудовлетворительно выполненный отчет также возвращается на доработку.

Отчет должен содержать заголовок, тему лабораторной работы, цель, задание, индивидуальную тему, описание хода выполнения работы, необходимые прикладные материалы (схемы, макеты документов и т.п.), в соответствии с требованиями к содержанию, и выводы по работе.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

- при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;
- при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;
- при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4.3. Методические указания по организации вычислительной инфраструктуры

Условия применения:

- Курс рассчитан на студентов 3-го года обучения.
- Наличие доступа к вычислительным ресурсам(GitLab).
- Разработаны ЛР, в них выделены части для проверки автотестами; инфраструктура для приёма задач(gitlab, CI/CD, autotests) согласована с ЛР, после завершения разработки автотестов ЛР переделаны и в них указаны требования к спецификации методов и коммитов.

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

- дать начальное представление о работе в IT инфраструктуре (приучить пользоваться гитом, приучить к требованиям к качеству разрабатываемого ПО на уровне прохождения тестов).

Задачи преподавателя:

- создание учетных записей студентов в gitlab вуза;
- GitLab Runner
- Работа над шаблонным репозиторием лабораторных работ;
- Автотесты ЛР1;
- Визуализация результатов тестирования;
- Написание инструкции для студентов;
- переработка ЛР 1 по итогам написания автотестов;
- написание автотестов ЛР N и встраивание их в общую инфраструктуру;
- переработка ЛР N по итогам написания автотестов;

Ожидаемые результаты студентов:

- начальное представление о работе в IT инфраструктуре (гит, нейминг, автотесты).

Порядок реализации

Задача №1: создание учетных записей студентов в gitlab вуза

Задача № 2: GitLab Runner.

Для выполнения CI/CD пайплайна и запуска автотестов был настроен GitLab Runner на удалённой виртуальной машине с ОС Ubuntu 24.04.

Последовательность настройки включала следующие шаги:

- Настройка системы – установка необходимых компонентов, таких, как Docker.
- Установка GitLab Runner по официальной инструкции.
- Регистрация Runner для частного сервера GitLab.
- Создание cron-скрипта для периодической очистки артефактов тестирования.

Задача №3: Работа над шаблонным репозиторием лабораторных работ

Этот репозиторий служит основой для всех новых лабораторных заданий и содержит преднастроенную структуру проекта, тестовую инфраструктуру и CI-конфигурации.

Ключевые файлы и их назначение:

`.gitlab-ci.yml` — основной конфигурационный файл CI/CD. Включает стадии `test`, `push_report`. Определяет среду выполнения (образ Docker), команды для запуска тестов и сборки.

`get_path.py` – скрипт для определения того, какие тесты использовать, по названию коммита.

`Dockerfile` – файл для запуска докер-образа с тестами.

`README.md` – инструкции к использованию шаблонного репозитория и тестов.

Также была реализована генерация отчётов о результатах тестирования кода. Вся структура максимально адаптирована для копирования студентами и минимизации порога входа при выполнении лабораторных

Задача №4: Визуализация результатов тестирования

Результаты тестирования представляются в виде html страницы, на которой можно посмотреть успешные тесты, ошибки и предупреждения.

Generated on Jul 22, 2025 at 21:31 Russian Standard Time

Test Results

Finished in 0.00s, 26613.23 tests/s, 70133.69 assertions/s

85 tests, 224 assertions, 0 failures, 0 errors, 0 skips

100% passed

Проверка количества публичных методов класса и объекта Student	2 tests, 2 assertions, 0 failures, 0 errors, 0 skips, finished in 0.0001s
✓ Количество методов класса	Assertions 1, time 0.000080s
✓ Количество методов объекта	Assertions 1, time 0.000028s
Проверка количества публичных методов класса и объекта StudentShort	2 tests, 2 assertions, 0 failures, 0 errors, 0 skips, finished in 0.0000s
✓ Количество методов класса	Assertions 1, time 0.000029s
✓ Количество методов объекта	Assertions 1, time 0.000021s
Тест альтернативного конструктора StudentShort.from_student	1 tests, 4 assertions, 0 failures, 0 errors, 0 skips, finished in 0.0000s
✓ Создание объекта StudentShort через альтернативный конструктор	Assertions 4, time 0.000027s
Тест инициализации объекта класса Student, геттеров и сеттеров	21 tests, 56 assertions, 0 failures, 0 errors, 0 skips, finished in 0.0003s
✓ test_telegram_setter	Assertions 2, time 0.000015s
✓ Геттер для email	Assertions 2, time 0.000008s
✓ Геттер для git	Assertions 2, time 0.000007s
✓ Геттер для id	Assertions 2, time 0.000008s
✓ Геттер для telegram	Assertions 2, time 0.000010s
✓ Геттер для имени	Assertions 2, time 0.000009s
✓ Геттер для отчества	Assertions 2, time 0.000018s
✓ Геттер для телефона	Assertions 2, time 0.000007s

Если метод написан некорректно, то будет выведена ошибка.

Порядок проверки корректности

Чек-лист для проверки лабораторных работ:

- гит инфраструктура с учетными записями студентов;
- шаблон гит репозитория для клонирования и работы студента с подключенными автотестами;
- инструкция по работе с гитом с подробным описанием именования методов и коммитов в ReadMe файле.

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Курс рассчитан на студентов 3-го года обучения.
- Наличие доступа к вычислительным ресурсам(GitLab).

- Разработана инфраструктура для приёма задач(gitlab, CI/CD, autotests) и согласована с лабораторными работами и настроена на всех студентов образовательной программы.

Цели, задачи и ожидаемые результаты

Цели организации лабораторных работ:

- Получение практического опыта ОО проектирования приложения.
- Получение практического опыта ОО проектирования предметной области
- Изучение ООП паттернов проектирования и архитектурных паттернов.
- Научиться строить описание архитектуры приложения с помощью диаграммы классов и диаграммы последовательностей
- Отработка умений реализации заданного архитектурного паттерна.
- Отработка навыков построения web-сервиса средствами JVM языков на примере фреймворков Spring, SpringBoot..

Задачи преподавателя:

- подготовка плана лабораторных работ;
- разработка примерной ЛР 1;
- организация git инфраструктуры для всех студентов на вычислительных мощностях ВУЗа и написание автотестов для примерного ЛР 1;
- переработка ЛР 1 по итогам написания автотестов;
- разработка примерного ЛР N;
- написание автотестов и встраивание их в общую инфраструктуру;
- переработка ЛР N по итогам написания автотестов;

Порядок реализации

Задача №1: Подготовка плана лабораторных работ (в соответствии с п.2.3.3 РПД)

Задача № 2: Разработка примерной ЛР, например ЛРН№4.

Факультет компьютерных технологий и прикладной математики
Кафедра вычислительных технологий
01.03.02

Паттерны проектирования
Лабораторная работа № 4. Классы модели

Цель – отработать принципы построения класса модели и аспекты реализации для разных хранилищ данных.

В рамках цели разрабатываются классы для работы с 5 разными хранилищами данных. Для достижения указанной цели необходимо самостоятельно реализовать класс для любого из хранилищ из заданий 1 или 2, а для остальных хранилищ воспользоваться AI генераторами кода . Обязательно указать, например, для СУБД Postgre написал класс самостоятельно, для остальных типов хранилищ воспользовался генератором. Важно понимать и ЗАПУСКАТЬ все реализованные методы. Важно реализовывать ровно указанные публичные методы и никаких больше. Важно понимать особенности реализации работы с разными хранилищами данных.

Каждое задание должно быть загружено на личный git-репозиторий отдельным коммитом. Коммиты должны иметь осмысленное название, при этом в названии содержать шифр лабы и задачи. Лабораторная работа выполняется в одной папке. Защита работы возможна на любой лабораторной работе от 6 до 16. Файлы должны иметь названия, соответствующие содержанию, а не нумерации заданий.

Если часть задач выполнена в один коммит, работа не проверяется. Если все коммиты сделаны в один час, работа не проверяется.

При защите обязательно писать скрипты работы с написанными классами по заданию преподавателя. На защите обязательно описать инструменты языка, позволяющие выполнять генерацию документации, публикацию библиотеки и логгирование событий программы.

При защите лабораторной работы в зависимости от уровня подготовки студента может быть необходимо дописать часть кода, тогда выставляется оценка ниже оценки 5. При защите может возникнуть ситуация, при которой студент не понимает, как пользоваться написанным инструментарием, в этом случае защита завершена, возобновится после полноценной подготовки студента и САМОСТОЯТЕЛЬНЫМ написанием всех методов класса заново.

(Зачтено) – любой из классов для работы с БД с логгированием и скриптами для установки ПО и библиотек.

Задание 1. StudentsListJSON, StudentsListYAML

1. Реализовать класс StudentsListJSON, который будет работать с текстовым файлом, содержащим списки студентов. Для этого необходимо обеспечить выполнение следующих функций:
 - a. Чтение всех значений из файла;
 - b. Запись всех значений в файл;
 - c. Получить объект класса Student по ID
 - d. getKNStudentShortList Получить список k по счету n объектов класса Student_short (например, вторые 20 элементов, чтобы в дальнейшем можно было листать длинный список), результат вернуть в формате DataList. Учесть в виде необязательного аргумента существующий объект класса DataList, вернуть или существующий, или измененный объект.
 - e. Сортировать элементы по набору ФамилияИнициалы.
 - f. Добавить объект класса Student в список (при добавлении сформировать новый ID).
 - g. Заменить элемент списка по ID.
 - h. Удалить элемент списка по ID.
 - i. get_student_short_count Получить количество элементов

2. Реализовать класс `StudentsListYAML`, который будет работать с текстовым файлом, содержащим списки студентов. Для этого необходимо обеспечить выполнение следующих функций:
 - a. Чтение всех значений из файла;
 - b. Запись всех значений в файл;
 - c. Получить объект класса `Student` по ID
 - d. `getKNStudentShortList` Получить список `k` по счету `n` объектов класса `Student_short` (например, вторые 20 элементов, чтобы дальше можно было листать длинный список), результат вернуть в формате `DataList`.
 - e. Сортировать элементы по набору `ФамилияИнициалы`.
 - f. Добавить объект класса `Student` в список (при добавлении сформировать новый ID).
 - g. Заменить элемент списка по ID.
 - h. Удалить элемент списка по ID.
 - i. `get_student_short_count` Получить количество элементов
3. Для класса `JSON` сгенерировать официальную документацию.
4. Класс `YAML` опубликовать в виде Java-библиотеки в `опенсорс`.
Продемонстрировать его использование в отдельном скрипте.

Задание 2 Подключение БД к модели. Работа осуществляется с 3 различными БД на выбор студентом.

1. Создать подключение.
2. Создать таблицу `student`. Для этого создать отдельную директорию, в которой будут храниться файлы, реализующие изменения в структуре БД. Создать два типа файлов, в одном – изменения в структуре, в другом – автоматические скрипты заполнения данными. Сохранить выполненные Вами скрипты в гит репозиторий.
3. Заполнить таблицу данными.
4. Протестируйте выполнение `select` запроса из этой таблицы из программы
5. Если не было сделано в предыдущем задании, то модифицируйте конструкторы или создайте альтернативные конструкторы для класса `Student` так, чтобы они принимали в качестве аргумента хеш, протестируйте написанные изменения.
6. Если необходимо, внесите изменения в уже существующие классы, чтобы весь написанный функционал продолжал работать.
7. Реализовать класс `StudentsListDB_1` (или 2 или 3 для каждой новой БД), который будет работать с созданной БД и таблицей, содержащей списки студентов. Для этого необходимо обеспечить выполнение следующих функций:
 - a. Получить объект класса `Student` по ID

- b. `getKNStudentShortList` Получить список `k` по счету `n` объектов класса `Student_short` (например, вторые 20 элементов, чтобы в дальнейшем можно было листать длинный список), результат вернуть в формате `DataList`.
 - c. Добавить объект класса `Student` в список (при добавлении сформировать новый ID).
 - d. Заменить элемент списка по ID.
 - e. Удалить элемент списка по ID.
 - f. Получить количество элементов
8. Подключить логгирование для каждого из трёх классов. Логги писать в отдельный файл, реализовать три разных уровня логгирования – от записи всех действий до записи исключительных ситуаций и ошибок.

Задание 3. Сформируйте отдельный скрипт, чтобы установить всё необходимое на Вашу ОС.

Задача №3: переработка ЛР по итогам написания автотестов

Для написанных автотестов вернуться к заданиям и указать точные спецификации файлов, методов проверки, добавить в ЛР инструкцию по работе в `git` университета и порядок сохранения и отображения результатов.

Порядок проверки корректности

Чек-лист для проверки лабораторных работ:

- ЛР1;
- гит инфраструктура с учетными записями студентов;
- шаблон гит репозитория для клонирования и работы студента с подключенными автотестами;
- инструкция по работе с гитом с подробным описанием именования методов и коммитов;
- автотесты для ЛР1;
- переделанное ЛР1;
- автотесты для ЛР1;
- переделанное ЛР1.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

5.1 Основная литература:

1) Оукс С. Эффективный Java. Тюнинг кода на Java 8, 11 и дальше. — 2-е межд. изд. — СПб.: Питер, 2023. — 480 с. — ISBN 978-5-4461-2345-6

2) Evans B., Verburg M. The Well-Grounded Java Developer. — 2nd ed. — Manning, 2024. — 520 с. — ISBN 978-1-61729-920-6

3) Рахимов Р. Java для опытных разработчиков. — 2-е изд. — СПб.: Питер, 2023. — 418 с. — ISBN 978-5-4461-1925-7

5.2 Дополнительная литература:

- 1 Документация Java.
- 2 Документация Java Spring.
- 3 Документация SpringBoot.

5.3. Периодические издания:

- 1 Базы данных компании «Ист Вью» <http://dlib.eastview.com>
- 2 Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

5.4. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

- 1 ЭБС «ЮРАЙТ» <https://urait.ru/>
- 2 ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» <http://www.biblioclub.ru/>
- 3 ЭБС «BOOK.ru» <https://www.book.ru>
- 4 ЭБС «ZNANIUM.COM» www.znanium.com
- 5 ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных

- 1 Scopus <http://www.scopus.com/>
- 2 ScienceDirect <https://www.sciencedirect.com/>
- 3 Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
- 4 Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
- 5 Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
- 6 Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
- 7 Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>
- 8 База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
- 9 Springer Journals: <https://link.springer.com/>
- 10 Springer Journals Archive: <https://link.springer.com/>
- 11 Nature Journals: <https://www.nature.com/>
- 12 Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
- 13 Springer Materials: <http://materials.springer.com/>
- 14 Nano Database: <https://nano.nature.com/>
- 15 Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
- 16 "Лекториум ТВ" <http://www.lektorium.tv/>
- 17 Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Бесплатные образовательные ресурсы

- 1 Jupyter Notebook – интерактивные вычисления
- 2 Visual Studio Code – редактор кода с поддержкой Python
- 3 Google Scholar/arXiv – доступ к научным публикациям

Ресурсы свободного доступа

- 1 КиберЛенинка <http://cyberleninka.ru/>;
- 2 Американская патентная база данных <http://www.uspto.gov/patft/>
- 3 Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
- 4 Федеральный портал "Российское образование" <http://www.edu.ru/>;

- 5 Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
- 6 Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
- 7 Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
- 8 Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
- 9 Служба тематических толковых словарей <http://www.glossary.ru/>;
- 10 Словари и энциклопедии <http://dic.academic.ru/>;
- 11 Образовательный портал "Учеба" <http://www.uceba.com/>;
- 12 Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--plai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

- 1 Электронный каталог Научной библиотеки КубГУ <http://megapro.kubsu.ru/MegaPro/Web>
- 2 Электронная библиотека трудов ученых КубГУ <http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
- 3 Среда модульного динамического обучения <http://moodle.kubsu.ru>
- 4 База учебных планов, учебно-методических комплексов, публикаций и конференций <http://infoneeds.kubsu.ru/>
- 5 Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru;>
- 6 Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
- 7 Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

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

По курсу предусмотрено проведение лекционных занятий, на которых дается систематизированный материал по java, паттернам проектирования, архитектурным паттернам. В ходе лекций рассматриваются ключевые концепции. После каждой лекции рекомендуется выполнение практических заданий для закрепления ключевых понятий и методов.

Лабораторные занятия курса посвящены практической разработке на Java. На занятиях студенты реализуют основные элементы архитектуры ООП приложения.

При самостоятельной работе студентам необходимо изучать рекомендованную литературу в виде официальной документации к используемым открытым программным продуктам, облачным платформам.

Важнейшим компонентом курса является самостоятельная проектная работа, в ходе которой команда студентов разрабатывает законченное решение задач (кейсов) промышленных партнеров.

Кейсы ПАО «Сбербанк»

1. Генеративный ИИ для автоматического составления инвестиционных обзоров

Описание:

Аналитики Сбера ежедневно составляют десятки аналитических и инвестиционных обзоров по рынкам, компаниям, макроэкономике. Задача — исследовать применение LLM для генерации кратких сводок и аналитических отчетов на основе входных данных: биржевые котировки, макроэкономические показатели, рыночные события.

Цель:

Разработать инструмент, способный по структурированным данным и краткому описанию формировать инвестиционный обзор в деловом стиле.

Ожидаемый результат:

Модель, генерирующая аналитические тексты длиной 500–1000 слов с разделами «обзор событий», «рекомендации», «прогнозы», оформленные в формате банка.

2. НЛП-анализ жалоб клиентов в свободной форме**Описание:**

В рамках клиентского сервиса Сбербанк обрабатывает обращения из чатов, мобильного приложения и жалобной формы. Требуется построить модель семантического анализа, выделяющую суть обращения, определяющую тональность и потенциальную серьёзность инцидента.

Цель:

Автоматизировать классификацию обращений для ускорения маршрутизации и выявления повторяющихся болевых точек в продуктах и процессах.

Ожидаемый результат:

Прототип модели, автоматически выделяющей темы жалоб (например, «ошибка в приложении», «двойное списание»), их эмоциональную окраску и критичность.

3. Генерация сценариев фишинговых писем для обучения сотрудников**Описание:**

Банк проводит киберучения, включая рассылку тестовых фишинговых писем сотрудникам для повышения их устойчивости к социальным атакам. Проект предполагает использование генеративной модели для создания реалистичных фишинговых писем различных типов (поддельные счета, HR-запросы, ИТ-поддержка).

Цель:

Создать генератор, способный на основе заданных параметров (тема, стиль, уровень угрозы) создавать тексты фишинга для тренировок.

Ожидаемый результат:

Набор разнообразных примеров фишинга и оценка их эффективности по реакции сотрудников, а также классификация моделей угроз.

4. Мультимодальный ассистент для банковских отделений**Описание:**

Физические отделения Сбербанка внедряют интерактивных консультантов.

Предполагается создание мультимодального ИИ-ассистента, который воспринимает речь и визуально ориентируется в пространстве (распознаёт клиента, документы, банкоматы), а также отвечает голосом.

Цель:

Разработать базовый прототип, имитирующий функциональность помощника: ответы на типовые запросы, визуальные подсказки, навигация по отделению.

Ожидаемый результат:

Интерактивная модель, объединяющая голосовой ввод, зрительное восприятие (например, QR-код паспорта), текстовый вывод и жестовую реакцию.

5. Объяснимость и контроль генеративных моделей в банковском ИИ**Описание:**

Банк активно использует LLM и NLP-сервисы (в чат-ботах, генерации шаблонов ответов, автоответах на e-mail), однако встает вопрос: как объяснять и контролировать поведение таких моделей, особенно в юридически значимых коммуникациях?

Цель:

Исследовать подходы к трассировке решений LLM (например, через логирование reasoning chain, пост-фильтрацию ответов, встроенные правила).

Ожидаемый результат:

Концепция системы explainability + compliance-модуля, обеспечивающего соответствие генерации стандартам банка и регулятора.

6. Генерация пользовательских сценариев работы в мобильном приложении**Описание:**

Банк хочет использовать генеративный ИИ для быстрой симуляции пользовательских сценариев — например, как клиент оформляет вклад, переводит средства, получает уведомление о риске мошенничества.

Цель:

Разработать генератор пошаговых сценариев пользовательского поведения с вариативностью (молодой клиент, пенсионер, ИП).

Ожидаемый результат:

Набор автоматически сгенерированных UX-сценариев, оформленных в виде сценариев для QA или UX-исследований, с логикой действий и типичными ошибками пользователя.

7. Генерация synthetic data для банковских моделей**Описание:**

Модели в Сбере требуют большого объёма транзакционных и клиентских данных, которые нельзя использовать напрямую из-за требований ЦБ и ФЗ-152. Задача — разработать метод генерации синтетических банковских данных, максимально близких к реальным по распределениям и поведению.

Цель:

Создать безопасный pipeline генерации данных (например, транзакций, профилей клиентов, шаблонов расходов) для обучения моделей.

Ожидаемый результат:

Синтетический датасет и отчет о метриках приближённости к реальному (TSNE, K-L divergence и др.), с оценкой пригодности для обучения скоринговых или антифрод-моделей.

8. Модель анализа инвестиционной привлекательности малого бизнеса**Описание:**

Банк активно развивает кредитование и инвестиционные инструменты для малого и среднего предпринимательства (МСП). Требуется создать модель, которая на основе открытых и банковских данных (выручка, расходы, тип деятельности, отзывы, онлайн-активность) оценивает инвестиционную привлекательность МСП.

Цель:

Разработать систему рейтинговой оценки компаний малого бизнеса с возможностью визуализации факторов и динамики показателей.

Ожидаемый результат:

Модель, присваивающая компании инвестиционный рейтинг (например, А–Е), объясняющая ключевые параметры и дающая рекомендации для инвестора.

9. Индивидуальная оценка кредитоспособности клиента на основе поведенческих данных**Описание:**

Современный кредитный скоринг выходит за рамки финансовых данных. Необходимо исследовать, как поведенческие и цифровые следы (частота входа в мобильный банк, способы оплаты, география, время отклика) влияют на персональную оценку риска.

Цель:

Разработать ML-модель, оценивающую вероятность дефолта по нестандартным поведенческим признакам (возможно — с explainable AI).

Ожидаемый результат:

Прототип скоринговой модели, которая, помимо стандартных данных, учитывает цифровой профиль клиента и объясняет решения (SHAP, LIME и др.).

10. Предиктивная аналитика возврата инвестиций по инфраструктурным проектам

Описание:

В ряде случаев Сбербанк выступает участником/инвестором в региональных инфраструктурных проектах (жилые массивы, дороги, технопарки). Задача — оценить прогнозируемую эффективность вложений с учётом демографии, миграции, экономической активности.

Цель:

Разработать модель, прогнозирующую ROI на горизонте 3–5 лет, используя внешние источники данных: Росстат, ЕГРЮЛ, кадастр, соцмедиа.

Ожидаемый результат:

Аналитическая модель с возможностью геовизуализации и сценарного анализа (рост/спад, госпрограммы, смена трафика и т.п.).

11. Анализ поведения пользователей в экосистеме цифрового рубля

Описание:

Сбербанк участвует в пилотных проектах по внедрению цифрового рубля. Интерес представляет исследование пользовательских паттернов: как изменяются модели потребления, скорости операций, уровень доверия, сравнение с классическим безналом.

Цель:

Построить модель анализа поведения клиентов, участвующих в транзакциях с цифровым рублем: частота, средний чек, контексты.

Ожидаемый результат:

Отчёт и ML-модель, классифицирующая типы пользователей и выявляющая ключевые различия в предпочтениях и барьерах цифровой валюты.

12. Сравнение text2video / text2img моделей

Описание:

Сбербанк заинтересован в сравнении text2video / text2img моделей (открытые модели, особенно китайские). Задача требует применения облачных ресурсов партнера для машинного обучения. От студентов требуется навык запуска открытых моделей, планирования, структурирования и логирования экспериментов, совместной работы. Задача может быть распараллелена для сравнения множества моделей независимо в группе студентов.

Цель:

Провести сравнение работы актуальных открытых моделей text2video / text2img.

Ожидаемый результат:

Таблица с результатами экспериментов модель / репозиторий / функционал / требования / оценка производительности / X примеров генераций (было/стало), human_eval по принципу арены (какая лучше)

Кейсы от «АВАЛАБ»

1. LLM и RAG для BI-системы Fastboard

Описание:

Для разрабатываемой компанией BI-системы Fastboard требуется разработать интерфейс на естественном языке для построения отчетов на больших массивах данных в ClickHouse. С помощью LLM необходимо классифицировать запросы пользователей на естественном языке и извлекать фактические параметры для дальнейшего вызова веб-сервиса отчетов.

Цель:

Разработать промпты для классификации и обработки запросов пользователей LLM и преобразования их к вызовам типовых отчетов с фактическими параметрами, извлекаемыми из запроса.

Ожидаемый результат:

Инструмент на основе LLM, позволяющий запрашивать данные о продажах.

2. Анализ обращений клиентов и CRM-переписки

Описание:

В службе клиентского сервиса застройщика ежедневно обрабатываются десятки обращений (e-mail, звонки, мессенджеры). Требуется реализовать систему семантического анализа и классификации NLU: выявлять суть обращений, уровень удовлетворенности, отслеживать повторяющиеся запросы.

Цель:

Автоматизировать первичный разбор и маршрутизацию запросов по тематике (сдача объекта, отделка, документы, жалоба и т.д.).

Ожидаемый результат:

Прототип, который выделяет суть обращений и формирует дашборд по текущим «болям» клиентов.

3. Генеративный ИИ для создания проектной документации по ТЗ

Описание:

В рамках проектирования объектов девелоперской компании архитекторы и инженеры тратят значительное время на подготовку текстовой проектной документации (обоснование решений, пояснительные записки, описания инженерных систем). Задача — исследовать возможность использования LLM для генерации черновиков проектной документации на основе исходных данных: этажность, материалы, климат, назначение, нормы.

Цель:

Разработать прототип текстового генератора, который помогает специалистам быстрее формировать документацию в соответствии с шаблонами и нормативами.

Ожидаемый результат:

Инструмент на основе LLM, создающий логически стройный и нормативно грамотный текст, поддающийся быстрой редакции инженером.

4. Мультимодальный агент для анализа строительных площадок

Описание:

ООО «АВА ЛАБ» разрабатывает систему для мониторинга строительных объектов. Требуется создать прототип мультимодального ИИ-агента, способного анализировать изображения со стройплощадки (видео/фото), а также принимать голосовые и текстовые запросы (например, «проверь монтаж перекрытия на 5 этаже»).

Цель:

Объединить возможности компьютерного зрения (распознавание стадии строительства, техники, нарушений) и НЛП (понимание запросов, отчетов).

Ожидаемый результат:

Интерактивный агент, который на запрос специалиста может показать нужный участок, прокомментировать прогресс, зафиксировать нарушения.

4. Генерация рекламного контента для жилых комплексов

Описание:

«АВА ГРУПП» регулярно запускает маркетинговые кампании для жилых комплексов. Необходимо исследовать использование диффузионных моделей для генерации изображений (визуализации интерьеров, окрестностей, видов из окон) и LLM — для описаний квартир, преимуществ района, инфраструктуры.

Цель:

Создать инструменты для быстрой генерации продающих материалов без привлечения дизайнеров и копирайтеров на первых этапах.

Ожидаемый результат:

Набор сгенерированных карточек объектов с текстом, изображением и логикой «живого» рекламного сообщения.

6. Генерация документации и шаблонов договоров

Описание:

Юридический департамент регулярно работает с договорами долевого участия, актами приёма-передачи и другими документами. Использование LLM может значительно сократить время на подготовку черновиков — достаточно ввести параметры сделки.

Цель:

Создать систему, которая генерирует адаптированные тексты документов по вводным данным (тип объекта, этаж, площадь, ФИО, сроки и пр.).

Ожидаемый результат:

Генератор документов в формате Word или PDF с автоматической подстановкой параметров и соблюдением юридического стиля.

7. Модель прогнозирования сроков сдачи объектов на основе текстовых и визуальных данных**Описание:**

Девелоперская компания ведёт аналитический архив по срокам строительства. С помощью мультимодальных моделей (текстовые отчёты + фото стройки) можно прогнозировать вероятность отклонения от графика сдачи.

Цель:

Разработать модель, которая по текущему статусу объекта (фото, отчёт СМР) оценивает риски задержек.

Ожидаемый результат:

Прототип, который показывает вероятность отклонений и даёт текстовые пояснения (основанные на распознанных признаках — «не завершены фасадные работы», «монтаж инженерии не начат»).

8. Обратная генерация — ИИ-помощник для покупателей квартир**Описание:**

Будущие покупатели часто задают типовые вопросы о квартирах, планировках, ипотеке, акциях, сроках. Вместо call-центра предлагается реализовать LLM-бота, который обрабатывает текстовые и голосовые запросы, показывает планировки, ссылается на PDF-документы и может «объяснять» информацию простым языком.

Цель:

Упростить коммуникацию с клиентами на этапе выбора квартиры и повысить качество первичного контакта.

Ожидаемый результат:

Демо-бот, способный отвечать на вопросы о жилом комплексе, ориентируясь в его характеристиках и маркетинговых документах.

Для студентов с ограниченными возможностями здоровья предусмотрены дополнительные индивидуальные консультации, на которых преподаватель подробно разъясняет сложные аспекты дисциплины, помогает адаптировать практические задания и обеспечивает специальные условия для освоения методов работы с системами искусственного интеллекта. Индивидуальный подход позволяет таким студентам полноценно участвовать в учебном процессе и достигать требуемых результатов обучения.

7. Материально-техническое обеспечение по дисциплине (модулю)**7.1 Перечень информационно-коммуникационных технологий****1. Облачные платформы и сервисы**

cloud.ru, YandexCloud, AWS/GCP/Azure – облачные вычисления

2. Системы управления версиями и коллаборации

Git/GitHub/GitLab – контроль версий кода и совместная разработка

4. Система управления обучением
Moodle – сдача работ

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

1. Свободное ПО (Open Source)
GitLab, GIT, MLFlow, Docker, Kubernetes, Terraform.

Виртуальные машины, кластер Managed Kubernetes и ресурсы GPU в облаке предоставляется индустриальным партнером ПАО «Сбербанк»:

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб

Дополнительные облачные ресурсы предоставляются технологическим партнером Yandex Cloud.

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.