

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.24 Объектно-ориентированное программирование

Направление подготовки 02.03.01 Математика и компьютерные науки

Профиль Искусственный интеллект и аналитика данных

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.01 Математика и компьютерные науки.

Программу составил(и):

А.И. Миков, профессор КИТ, д.ф.м.н., профессор
И.О. Фамилия, должность, ученая степень, ученое звание


_____ подпись

А.А. Миков, преподаватель КИТ
И.О. Фамилия, должность, ученая степень, ученое звание


_____ подпись

Рабочая программа дисциплины утверждена на заседании кафедры информационных технологий протокол № 1 от «26» августа 2025г.

Заведующий кафедрой Подколзин В.В.



Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 1 от «28» августа 2025г.

Председатель УМК факультета
А. В. Коваленко


_____ подпись

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 бюджетное образовательное учреждение 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 методологии объектно-ориентированного программирования.

1.2 Задачи дисциплины

1. Изучение подхода объектно-ориентированного программирования на примере языка C#.
2. Изучение основных принципов ООП и подходов к разработке объектной модели.
3. Изучение классических паттернов проектирования и, в частности, возможностей их применения в рефакторинге исходного кода.
4. Получение практического опыта в применении ООП к задачам ИИ.
5. Получение практического опыта в работе с современными библиотеками на платформе .NET.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование» относится к дисциплинам базовой части, код Б1.О.24.

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

1. Программирование
2. Алгоритмы и структуры данных

Требованием к «входным» знаниям является понимание основ программирования на императивных языках.

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: AI Architect (Архитектор ИИ)

Задачи:

- Проектирование архитектуры ИИ-систем
- Выбор технологий и инструментов для ИИ-проектов
- Интеграция ИИ-компонентов в корпоративные системы
- Обеспечение масштабируемости и производительности
- Техническое руководство ИИ-проектами

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** *Способен проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности*
- ОПК-2.1** Применяет современные математические и вычислительные методы для решения научных задач в рамках поставленной проблемы
Знает концепции, положенные в основу системного программного обеспечения .NET
Умеет применять математические методы для построения эффективных алгоритмов с учетом архитектуры компьютеров
Владеет основами формализации задач для объектно-ориентированного проектирования при разработке эффективных программных решений, создании программных продуктов
- ОПК-6** *Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения*
- ОПК-6.2** Применяет методы структурного и объектно-ориентированного программирования, создает модульные и масштабируемые программы
Знает основные конструкции объектно-ориентированного языка C#, необходимые для создания программных продуктов и программных комплексов различного назначения, в т.ч. для анализа данных
Умеет разрабатывать алгоритмы и объектно-ориентированные программы для анализа данных на языке C#, готовые к интеграции в реальные системы
Владеет современными информационными технологиями, в том числе базовыми методами объектно-ориентированного программирования на платформе .NET
- ОПК-7** *Способен использовать основы экономических знаний в различных сферах жизнедеятельности*
- ОПК-7.1** Анализирует экономическую эффективность IT-решений
Знает экономические концепции, положенные в основу системного программного обеспечения .NET
Умеет применять математические методы для построения экономически эффективных алгоритмов
Владеет основами формализации задач для объектно-ориентированного проектирования при разработке эффективных программных решений
- PL-2A** *Способен применять языки программирования платформы .NET для решения задач в области ИИ*
- PL-2A.1** Разрабатывает и отлаживает прикладные решения с элементами ИИ с использованием языка программирования C#
Уверенно владеет синтаксисом языка программирования C# и использует стандартную библиотеку .NET. Уверенно владеет механизмами обработки данных на языке программирования C# (LINQ, SciSharp, Math.NET), механизмами работы с асинхронным/параллельным кодом. Имеет общее понимание работы среды .NET (CLR, промежуточный язык, сборка мусора).

Имеет навыки разработки поддерживаемого программного кода с использованием ООП.

Имеет навыки разработки серверных приложений с использованием общепринятых фреймворков/библиотек (ASP.NET), навыки работы с базами данных как напрямую при помощи драйверов и соответствующих языков запросов, так и при помощи ORM-фреймворков.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 3 зач. ед. (108 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)				
		3				
Контактная работа, в том числе:	54,3	54,3				
Аудиторные занятия (всего):	50	50				
Занятия лекционного типа	16	16				
Лабораторные занятия	34	34				
Занятия семинарского типа (семинары, практические занятия)						
Иная контактная работа:	2,3	2,3				
Контроль самостоятельной работы (КСР)	2	2				
Промежуточная аттестация (ИКР)	0,3	0,3				
Самостоятельная работа, в том числе:	20	20				
Курсовая работа						
Выполнение индивидуальных заданий	20	20				
Реферат						
Подготовка к текущему контролю						
Контроль:	35,7	35,7				
Подготовка к экзамену	35,7	35,7				
Общая трудоемкость	час.	108	108			
	в том числе контактная работа	54,3	54,3			
	зач. ед	3	3			

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 3 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Быстрый старт - основы, математические параллели	12	2		6	4
2.	Классы и объекты в разных языках программирования	12	2		6	4
3.	"Под капотом". Эффективность и проблемы	10	2		4	4

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
4.	Паттерны объектно-ориентированного проектирования	20	6		10	4
5.	Современные тенденции. ООП и ИИ. ООП и GameDev	16	4		8	4
ИТОГО по разделам дисциплины		70	16		34	20
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		0,3				
Подготовка к текущему контролю		35,7				
Общая трудоемкость по дисциплине		108				

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.	Быстрый старт - основы, математические параллели	Программирование как взгляд на мир. Объектная структура цифрового мира. Программирование и современная математика. Системный подход к программированию. Н.Вирт: "Алгоритмы + структуры данных = программы". Наука программирования = Computer Science. Класс как множество объектов. Вложение множеств: наследование. Интерфейсы. Влияние алгебраических структур. Примеры простых программ на C#. Характеристика языка и платформы .NET. Полиморфизм - влияние математической теории категорий и функторов. ООП и другие парадигмы программирования, сравнительный анализ.	ЛР
2.	Классы и объекты в разных языках программирования	Объявление классов и интерфейсов. Методы. Абстрактные классы. Обобщения (Generics). Создание экземпляров. Модификаторы доступа, их назначение. Ключевые принципы ООП: наследование, инкапсуляция, полиморфизм, (передача данных).	ЛР
3.	"Под капотом". Эффективность и проблемы	Ссылочные и значимые типы, как они хранятся в памяти и как используются. Сборщик мусора, принципы работы.	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		Исполнение программы, как исходный код исполняется процессором, CLR, JVM.	
4.	Паттерны объектно-ориентированного проектирования	Что такое грязный и чистый код? SOLID. Структурные, поведенческие и порождающие паттерны. Классические примеры из «Приёмы объектно-ориентированного проектирования. Паттерны проектирования» Gang Of Four.	ЛР
5.	Современные тенденции. ООП и ИИ. ООП и GameDev	Яркий пример работы с ООП - видеоигры. Паттерны в видеоиграх: Singleton, StateMachine, ObjectPool Альтернативный подход: DOD и ECS. Использование ООП в ИИ. Агенты, Behaviours, SemanticKernel.	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.	Быстрый старт - основы, математические параллели	1. Настройка рабочего окружения.	ЛР
2.		2. Работа со стандартным синтаксисом.	ЛР
3.		3. Стандартные структуры данных.	ЛР
4.	Классы и объекты в разных языках программирования	4. Наследование и полиморфизм. Интерфейсы и абстрактные классы.	ЛР
5.		5. Работа с данными.	ЛР
6.		6. Исключения и их обработка.	ЛР
7.	"Под капотом". Эффективность и проблемы	7. Обобщения. Коллекции.	ЛР
8.		8. IDisposable, GC.	
9.	Паттерны объектно-ориентированного проектирования	9. Рефакторинг существующих решений.	ЛР
10.		10. Порождающие паттерны.	ЛР
11.		11. Структурные паттерны.	ЛР
12.		12. Поведенческие паттерны.	ЛР
13.		13. Оценка применимости паттернов к решаемой задаче.	ЛР
14.	Современные тенденции. ООП и ИИ. ООП и GameDev	14.SemanticKernel. Основы.	ЛР
15.		15. SemanticKernel. Плагины, MCP.	ЛР
16.		16. Классические паттерны в системах со сложной логикой и акцентом на производительность.	ЛР
17.		17. Data Oriented Design.	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 информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

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 образовательные технологии: 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.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
3	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	34
Итого			34

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 «Объектно-ориентированное программирование».

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

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.	Быстрый старт - основы, математические параллели	PL-2A.1	ЛР 1-3	Вопросы к экзамену 1 - 6
2.	Классы и объекты в разных языках программирования	PL-2A.1	ЛР 4-6	Вопросы к экзамену 7 - 12
3.	"Под капотом". Эффективность и проблемы	PL-2A.1; ОПК-7.1	ЛР 7-8	Вопросы к экзамену 13 - 18
4.	Паттерны объектно-ориентированного проектирования	ОПК-2.1	ЛР 9-13	Вопросы к экзамену 19 - 24
5.	Современные тенденции. ООП и ИИ. ООП и GameDev	ОПК-6.2	ЛР 14-17	Вопросы к экзамену 25 - 30

Показатели, критерии и шкала оценки сформированных компетенций

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

ОПК-2 *Способен проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности*

- ОПК-2.1** Применяет современные математические и вычислительные методы для решения научных задач в рамках поставленной проблемы
Знает концепции, положенные в основу системного программного обеспечения .NET
Умеет применять математические методы для построения эффективных алгоритмов с учетом архитектуры компьютеров
Владеет основами формализации задач для объектно-ориентированного проектирования при разработке эффективных программных решений, создании программных продуктов
- ОПК-6** *Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения*
- ОПК-6.2** Применяет методы структурного и объектно-ориентированного программирования, создает модульные и масштабируемые программы
Знает основные конструкции объектно-ориентированного языка C#, необходимые для создания программных продуктов и программных комплексов различного назначения, в т.ч. для анализа данных
Умеет разрабатывать алгоритмы и объектно-ориентированные программы для анализа данных на языке C#, готовые к интеграции в реальные системы
Владеет современными информационными технологиями, в том числе базовыми методами объектно-ориентированного программирования на платформе .NET
- ОПК-7** *Способен использовать основы экономических знаний в различных сферах жизнедеятельности*
- ОПК-7.1** Анализирует экономическую эффективность IT-решений
Знает экономические концепции, положенные в основу системного программного обеспечения .NET
Умеет применять математические методы для построения экономически эффективных алгоритмов
Владеет основами формализации задач для объектно-ориентированного проектирования при разработке эффективных программных решений
- PL-2A** *Способен применять языки программирования платформы .NET для решения задач в области ИИ*
- PL-2A.1** Разрабатывает и отлаживает прикладные решения с элементами ИИ с использованием языка программирования C#
Уверенно владеет синтаксисом языка программирования C# и использует стандартную библиотеку .NET. Уверенно владеет механизмами обработки данных на языке программирования C# (LINQ, SciSharp, Math.NET), механизмами работы с асинхронным/параллельным кодом. Имеет общее понимание работы среды .NET (CLR, промежуточный язык, сборка мусора). Имеет навыки разработки поддерживаемого программного кода с использованием ООП.
Имеет навыки разработки серверных приложений с использованием общепринятых фреймворков/библиотек (ASP.NET), навыки работы с базами данных как напрямую при помощи драйверов и соответствующих языков запросов, так и при помощи ORM-фреймворков.

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

- ОПК-2** *Способен проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности*
- ОПК-2.1** Применяет современные математические и вычислительные методы для решения научных задач в рамках поставленной проблемы
Знает концепции, в т.ч. экономические, положенные в основу системного программного обеспечения .NET
Умеет применять математические методы для построения экономически эффективных алгоритмов с учетом архитектуры компьютеров и сетей
Владеет основами формализации задач для объектно-ориентированного проектирования при разработке эффективных программных решений, создании и сопровождении программных продуктов и программных комплексов
- ОПК-6** *Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения*
- ОПК-6.2** Применяет методы структурного и объектно-ориентированного программирования, создает модульные и масштабируемые программы
Знает синтаксис и семантику объектно-ориентированного языка C#, используемого для создания программных продуктов и программных комплексов различного назначения, в т.ч. для анализа данных
Умеет разрабатывать алгоритмы и объектно-ориентированные программы для анализа данных на языке C#, соответствующие стандартам качества и готовые к интеграции в реальные системы
Владеет современными информационными технологиями при создании программных продуктов и программных комплексов различного назначения, в том числе базовыми методами объектно-ориентированного программирования на платформе .NET
- ОПК-7** *Способен использовать основы экономических знаний в различных сферах жизнедеятельности*
- ОПК-7.1** Анализирует экономическую эффективность IT-решений
Знает экономические концепции, положенные в основу системного программного обеспечения .NET
Умеет применять математические методы для построения экономически эффективных алгоритмов с учетом архитектуры компьютеров и сетей
Владеет основами формализации задач для объектно-ориентированного проектирования при разработке эффективных программных решений, создании программных продуктов и программных комплексов
- PL-2A** *Способен применять языки программирования платформы .NET для решения задач в области ИИ*
- PL-2A.1** Разрабатывает и отлаживает прикладные решения с элементами ИИ с использованием языка программирования C#
Уверенно владеет продвинутыми инструментами оптимизации работы с данными на языке программирования C# (stackalloc, Array/ObjectPool, MemoryMarshal), инструментами профилирования программного кода C# (BenchmarkDotNet, dotTrace/dotMemory).
Имеет навыки проектирования и разработки реактивных клиент-серверных приложений интегрирующих ИИ-технологии.
Имеет навыки проектирования и реализации масштабируемых систем, взаимодействующих с инструментами ИИ. Умеет проектировать масштабируемую программную архитектуру для реализации систем взаимодействия с пользователем с использованием ИИ.

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

- ОПК-2** *Способен проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности*
- ОПК-2.1** Применяет современные математические и вычислительные методы для решения научных задач в рамках поставленной проблемы
Знает концепции, положенные в основу системного программного обеспечения .NET, а также концепцию паттернов объектно-ориентированного программирования
Умеет применять математические и вычислительные методы для построения эффективных алгоритмов и структур данных с учетом архитектуры компьютеров и сетей различных масштабов
Владеет основами формализации прикладных и научных задач для объектно-ориентированного проектирования при разработке эффективных программных решений, создании и сопровождении программных продуктов и программных комплексов различного назначения
- ОПК-6** *Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения*
- ОПК-6.2** Применяет методы структурного и объектно-ориентированного программирования, создает модульные и масштабируемые программы
Знает синтаксис, семантику и прагматику объектно-ориентированного языка C#, используемого для создания программных продуктов и программных комплексов различного назначения, в т.ч. для анализа данных и решения задач искусственного интеллекта
Умеет разрабатывать алгоритмы и объектно-ориентированные программы для анализа данных на языке C#, соответствующие стандартам качества и готовые к интеграции в реальные системы; оценивать сложность алгоритмов
Владеет современными информационными технологиями при создании программных продуктов и программных комплексов различного назначения, в том числе методами объектно-ориентированного программирования на платформе .NET; методами разработки модульных и масштабируемых программ
- ОПК-7** *Способен использовать основы экономических знаний в различных сферах жизнедеятельности*
- ОПК-7.1** Анализирует экономическую эффективность IT-решений
Знает экономические концепции, положенные в основу системного программного обеспечения .NET, а также концепцию паттернов объектно-ориентированного программирования
Умеет применять математические и вычислительные методы для построения экономически эффективных алгоритмов и структур данных с учетом архитектуры компьютеров и сетей различных масштабов
Владеет основами формализации прикладных задач для объектно-ориентированного проектирования при разработке экономически эффективных программных решений
- PL-2A** *Способен применять языки программирования платформы .NET для решения задач в области ИИ*
- PL-2A.1** Разрабатывает и отлаживает прикладные решения с элементами ИИ с использованием языка программирования C#
Уверенно владеет навыками проектирования клиент-серверных систем использующих ИИ, умеет проектировать полный цикл разработки

приложений, имеет навыки развертывания приложений в облачной инфраструктуре, автоматизации процессов тестирования/сборки/развертывания.

Обладает навыками продвинутой работы с системами сборки программных пакетов на языке программирования C# (MSBuild, NuGet). Знаком с настройкой и эксплуатацией инструментов контроля качества кода (editorconfig, dotnet format).

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

Практические задания для лабораторных работ

Изучение стандартных CLI команд, настройка проекта, объявление зависимостей и внешних пакетов.

Анализ предметной области с извлечением иерархии классов.

Реализация класса, вычисляющего значения функции, представленные обратной польской записью.

Конвертация императивной программы в объектно-ориентированную.

Обработка ошибок ввода данных (деление на ноль, некорректный формат) с использованием try-catch и кастомных сообщений.

Разработка универсального хранилища данных (например, «ящик» для книг и инструментов) с применением обобщений (Generics).

Работа с текстовым файлом: запись и чтение данных с автоматическим освобождением ресурсов через оператор using.

Разделение монолитного класса «Автомобиль» (двигатель, трансмиссия) на отдельные компоненты с инкапсуляцией логики.

Создание фабрики для генерации животных (кошка, собака) с выводом звуков, используя паттерн «Простая фабрика».

Моделирование системы уведомлений (например, прогноз погоды) с подпиской на события через паттерн «Наблюдатель».

Пример лабораторной работы ***Лабораторная работа №9: рефакторинг.*** **Тема: Паттерны ООП.**

Цель работы:

Находить минусы существующих реализаций и обновлять их для создания более гибкой кодовой базы.

Задачи:

Изучить задачу, решаемую в рамках конкретной реализации, найти минусы и определить как можно переделать существующую реализацию.

Ожидаемые результаты:

Эмпирически более гибкое решение.

Ход работы

1. Изучение решаемой задачи
2. Изучение существующей кодовой базы

3. Подбор нового решения
4. Реализация нового решения и проверка на тестовых кейсах

Требования к отчету

1. Титульный лист

Название работы, ФИО студента, группа, дата.

2. Введение

Цель и задачи работы.

Описание задачи.

3. Теоретическая часть

Кратко описать суть работы. Пояснить слабые места существующей реализации, как ее можно улучшить.

4. Реализация

Исходный код.

5. Результаты

Описание исходного кода, результаты прохождения тест-кейсов.

6. Выводы

Проблемы, возникшие при выполнении задания.

Критерии оценки

Зачтено: убедительные доводы о проблемах, успешное прохождение тест-кейсов.

Не зачтено: исходный код не работает

Рекомендуемые инструменты

- GitLab, GIT.

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

Вопросы для подготовки к экзамену

1. Дайте определение объектно-ориентированного программирования и перечислите его основные принципы.
2. Что такое класс и объект в контексте C#?
3. Объясните различие между ссылочными и значимыми типами данных.
4. В чём заключается принцип инкапсуляции и как он реализуется в C#?
5. Что представляет собой наследование и как оно выражается в синтаксисе языка?
6. Как реализуется полиморфизм в C#? Приведите пример.
7. Какова роль интерфейсов в ООП и чем они отличаются от абстрактных классов?
8. Что делает сборщик мусора (Garbage Collector) и какие объекты он освобождает?
9. Какова цель модификаторов доступа? Назовите основные модификаторы в C#.
10. Что такое исключение (exception) и как осуществляется его обработка?
11. Что такое конструктор и деструктор? Какую роль они играют в жизненном цикле объекта?
12. В чём различие между перегрузкой и переопределением методов?

13. Объясните влияние алгебраических структур и теории множеств на понятие класса и наследования.
14. Как принципы SOLID повышают качество объектно-ориентированного кода?
15. Приведите примеры порождающих паттернов и объясните, в каких случаях их целесообразно применять.
16. Объясните отличие структурных и поведенческих паттернов проектирования.
17. Какие ошибки проектирования нарушают принцип единственной ответственности (SRP)?
18. Как обобщения используются при проектировании коллекций и алгоритмов?
19. Какие преимущества даёт использование паттерна Factory Method по сравнению с прямым созданием объектов?
20. Как можно использовать делегаты и события для реализации принципов инверсии управления (IoC)?
21. Как подход Data-Oriented Design (DOD) противопоставляется ООП?
22. Приведите пример реализации паттерна Singleton с учётом требований к многопоточности.
23. Как оценить компромисс между производительностью и чистотой архитектуры при использовании ООП?
24. Какие проблемы масштабируемости возникают в ООП-проектах и как их можно минимизировать с помощью паттернов?
25. Паттерны GoF и современность: Почему классические паттерны (например, Singleton) считаются анти-паттернами в контексте распределенных систем?
26. Чистота кода и технический долг: Как метрики сложности (циклическая сложность, глубина наследования) коррелируют с затратами на поддержку?
27. Паттерн State Machine в ИИ: Как конечные автоматы используются для реализации поведения NPC? В чем ограничения этого подхода при создании адаптивного ИИ?
28. Рефакторинг легаси-кода: Какие шаги предпринимаются при рефакторинге монолитного класса с сотней методов? Как паттерны «Стратегия» и «Фасад» помогают изолировать изменения?
29. Исключения и отказоустойчивость: Почему в высоконагруженных системах исключения заменяют на Result-типы (как в Rust)? Как это влияет на дизайн API?
30. Коллекции и алгоритмы: Как выбор коллекции (List vs Dictionary vs HashSet) влияет на асимптотическую сложность операций в реальных задачах (кэширование, поиск)?

Перечень компетенций (части компетенции), проверяемых оценочным средством:
PL-2A.1 (вопросы 1 – 18); ОПК-2.1 (вопросы 19 – 25); ОПК-6.2 (вопросы 25 – 30); ОПК-7.1 (вопросы 23 – 26).

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 экзамена: устно.

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

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

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и ответов на вопросы экзамена.

Критерии оценки:

оценка «неудовлетворительно» выставляется в случае выполнения одного из условий:

- непонимание сущности излагаемых вопросов, грубые ошибки в ответе, неуверенные и неточные ответы на дополнительные вопросы;
- выполнено менее 50% контрольных заданий.

оценка «удовлетворительно» в случае выполнения условий:

- частично ответил на два вопроса экзаменационного билета или достаточно полно ответил хотя бы на один вопрос;
- выполнено не менее 50% контрольных заданий.

оценка «хорошо» в случае выполнения условий:

- достаточно полно ответил на два вопроса экзаменационного билета;
- даны частичные ответы на дополнительные вопросы;
- выполнено не менее 60% контрольных заданий.

оценка «отлично» в случае выполнения условий:

- глубокие исчерпывающие знания по вопросам экзаменационного билета;
- даны правильные и конкретные ответы на дополнительные вопросы;
- сданы все тесты и выполнено не менее 80% контрольных заданий.

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 1 года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab, компиляторы C#).
- Индивидуальные задания включают задачи на реализацию алгоритмов и структур данных, проверяемые автотестами.

Цели, задачи и ожидаемые результаты:

Цели:

- Обеспечить студентов инструментами для работы с алгоритмами и структурами данных.
- Приучить к использованию Git, CI/CD и автотестов для контроля качества кода.

Задачи преподавателя:

1. Создание учетных записей студентов в GitLab.
2. Настройка GitLab Runner для автоматического тестирования.
3. Разработка шаблонного репозитория для лабораторных работ.
4. Написание автотестов для проверки корректности реализации алгоритмов.
5. Визуализация результатов тестирования.

Ожидаемые результаты студентов:

- Умение работать с Git и CI/CD.
- Навыки написания чистого, тестируемого кода на C#.
- Понимание важности автоматической проверки алгоритмов.

Порядок реализации:

1. Создание учетных записей в GitLab:

Каждый студент получает доступ к приватному репозиторию.

2. Настройка GitLab Runner:

Используется Docker-образ с компилятором C# (g#/clang) и фреймворками для тестирования.

3. Шаблонный репозиторий:

Включает:

- `.gitlab-ci.yml` для автоматического тестирования.

- README .md с инструкциями.

- Примеры кода и тестов.

4. Автотесты:

Проверяют корректность реализации алгоритмов (например, сортировки, работы с деревьями, хеш-таблицами).

5. Визуализация результатов:

Генерация отчетов с указанием успешных и проваленных тестов.

Порядок проверки корректности:

Чек-лист:

- Наличие Git-репозитория у всех студентов.
- Корректная работа CI/CD.
- Автотесты покрывают ключевые функции.

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Курс включает 9 лабораторных работ (см. раздел 2.3.3 РПД).

- Для выполнения требуется:

Среда разработки (Visual Studio).

GitLab для сдачи заданий.

Цели, задачи и ожидаемые результаты:

Цели:

- Практическое освоение алгоритмов и структур данных.
- Развитие навыков отладки и оптимизации кода.

Задачи преподавателя:

1. Подготовка плана лабораторных работ.
2. Разработка индивидуальных заданий.
3. Организация Git-инфраструктуры и автотестов.

Ожидаемые результаты студентов:

- Умение реализовывать и анализировать алгоритмы.
- Навыки работы с Git и CI/CD.

Порядок реализации:

1. План лабораторных работ:

Соответствует темам из РПД (сортировки, деревья, графы, хеш-таблицы и т. д.).

2. Пример задания (ЛР №3: "Сравнение алгоритмов сортировки"):

- Реализовать сортировки: пузырьком, выбором, быструю.
- Сравнить время работы на разных входных данных.
- Автотесты проверяют корректность сортировки и замеряют время.

3. Критерии оценки:

- **зачтено:** Полная реализация, анализ сложности, код проходит тесты.
- **Незачтено:** Код не работает или не соответствует ТЗ.

Порядок проверки корректности:

Чек-лист:

- Наличие автотестов для каждой лабораторной работы.
- Инструкции по именованию коммитов и методов.

4.5. Методические указания по организации проектной деятельности студентов

Условия применения:

- Курс включает проектную работу (16 часов).
- Доступны кейсы от преподавателей.

Цели, задачи и ожидаемые результаты:

Цели:

- Применение алгоритмов и структур данных в реальных задачах.
- Развитие навыков командной работы.

Задачи преподавателя:

1. Сбор кейсов (например, оптимизация поиска в больших данных).
2. Формирование ТЗ для проектов.
3. Разработка системы оценки.

Ожидаемые результаты студентов:

Умение решать прикладные задачи с использованием изученных алгоритмов.

Порядок реализации:

1. Пример кейса:

Оптимизация хеш-таблицы для ускорения поиска в базе данных.

2. ТЗ для проекта:

- Реализовать хеш-таблицу с методами цепочек и открытой адресации.
- Сравнить производительность на реальных данных.

3. Критерии оценки:

- Корректность реализации.
- Анализ эффективности (O-нотация).

Порядок проверки корректности:

Чек-лист:

- Наличие минимум 10 кейсов.
- Четкие критерии оценки проектов.

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1 Учебная литература:

1) Прайс М., Гандерлой Дж. Visual C# .NET. Полное руководство. — М.: ДМК, 2023. — 960 с. — ISBN 978-5-7931-0490-6

2) Фримен Э., Робсон Э. Паттерны проектирования для C# и .NET Core. — СПб.: Питер, 2024. — 480 с. — ISBN 978-5-4461-2346-3

3) Скит Дж. C# для профессионалов. — 5-е изд. — СПб.: Питер, 2025. — 600 с. — ISBN 978-5-4461-1927-1

- Документация GitLab.

5.2. Периодические издания:

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

5.3. 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.prlib.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. Visual Studio Code – редактор кода
2. 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/>

5.5 Публикации конференций А*

1. Höller, D., Behnke, G., Bercher, P., Biundo-Stephan, S., Fiorino, H., Pellier, D., & Alford, R. (2020). HDDL: An Extension to PDDL for Expressing Hierarchical Planning Problems. AAAI Conference on Artificial Intelligence.
2. Feng, F., & Magliacane, S. (2023). Learning Dynamic Attribute-factored World Models for Efficient Multi-object Reinforcement Learning. Neural Information Processing Systems, abs/2307.09205.
3. Zan, D., Chen, B., Yang, D., Lin, Z., Kim, M., Guan, B., Wang, Y., Chen, W., & Lou, J. (2022). CERT: Continual Pre-Training on Sketches for Library-Oriented Code Generation. International Joint Conference on Artificial Intelligence, abs/2206.06888.
4. Cipriano, B.P., & Alves, P. (2024). LLMs Still Can't Avoid Instanceof: An Investigation Into GPT-3.5, GPT-4 and Bard's Capacity to Handle Object-Oriented Programming Assignments. 2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), 162-169.

5.4 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

5.5 Перечень лицензионного и свободно распространяемого программного обеспечения

1. Свободное ПО (Open Source)
GitLab, GIT, Visual Studio Code.
2. Лицензионное ПО
Visual Studio, JetBrains Rider, SemanticKernel

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

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

Лабораторные занятия курса посвящены практическому освоению методов объектно-ориентированного программирования. В ходе занятий разбираются готовые программные приложения и проводится анализ их построения. После занятия

рекомендуется выполнить упражнения, приводимые в аудитории для самостоятельной работы.

При самостоятельной работе студентов необходимо изучить литературу, приведенную в перечнях выше, для осмысления вводимых понятий, анализа предложенных подходов и методов разработки программ. Разрабатывая решение новой задачи, студент должен уметь выбрать эффективные и надежные структуры данных для представления информации, подобрать соответствующие алгоритмы для их обработки, учесть специфику языка программирования, на котором будет выполнена реализация. Студент должен уметь выполнять тестирование и отладку алгоритмов решения задач с целью обнаружения и устранения в них ошибок.

Важнейшим этапом курса является самостоятельная работа по дисциплине. В процессе самостоятельной работы студент приобретает навыки объектно-ориентированного программирования.

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. NLP-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. Мультиmodalный ассистент для банковских отделений

Описание:

Физические отделения Сбербанка внедряют интерактивных консультантов. Предполагается создание мультиmodalного 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, отчётов).

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.

Кейс от ООО «СвязьРесурс-Кубань»**Описание:**

Компания ООО "СвязьРесурс-Кубань" оказывает услуги связи. Работа с клиентами автоматизирована на базе CRM Битрикс 24. Для компании актуальны вопросы разработки первоначальных версий документов с помощью LLM и в перспективе автоматизации генерации большого количества документов по шаблонам с помощью LLM и RAG системы с интеграцией с Битрикс 24. Задачи включают в себя:

1. Разработка библиотеки промптов для генерации регламентов описания бизнес-процессов Битрикс 24.
2. Разработка библиотеки промптов для генерации техзаданий на основе параметров оказания услуг.
3. Разработка библиотеки промптов для генерации коммерческих предложений на основе параметров оказания услуг.
4. Разработка библиотеки промптов для генерации скриптов работы технической поддержки.
5. Разработка библиотеки промптов для генерации скриптов работы отдела продаж.
6. Апробация и сравнение различных языковых моделей для решения задач.

Цель:

Автоматизировать работу сотрудников по составлению типовых документов.

Ожидаемый результат:

Библиотека промптов и рекомендации по использованию LLM для решения поставленных задач.

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)

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
2	Виртуальная машина с GPU	Аренда публичного IP	1		Шт
		Виртуальная машина с GPU NVIDIA® Tesla® V100 2 GPU 8 vCPU 128 ГБ RAM	1	1	Шт
		ОС Ubuntu_24.04	1		Шт
		Системный диск SSD	1		Шт
			2000		Гб
		Диск SSD	1		Шт
			4096		Гб
		Диск SSD	1		Шт
			4096		Гб
			Аренда публичного IP	1	
3	K8S	Master node 8 vCPU 16 RAM	1	1	Шт
		Worker node 10% доля 4 vCPU 32 RAM	5		Шт
		Worker node SSD-NVME	64		Гб
		Аренда публичного IP	1		Шт
4	ML Inference Instance Type GPU	Время работы в месяц	40	1	Ч
		Инстанс 8 x NVIDIA® H100 NVLink PCIe 160 vCPU 1520 GB RAM	1		Шт
		Количество запросов к ML-моделям	1		Млн. Шт
		Кэш ML-моделей	160		Гб
5	LLM	Токены GigaChat 2 Max	50		Млн. Шт
		Токены Embeddings	400		Млн. Шт

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.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
4.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	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.