

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.

« 29 » августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
Б1. В.01 Мультиагентные системы

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 «Мультиагентные системы» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.03.02 Прикладная математика и информатика.

Программу составил(и):

С. Г. Сеница, доцент КИТ, к.т.н.

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



подпись

Рабочая программа дисциплины «Мультиагентные системы» утверждена на заседании центра искусственного интеллекта протокол № 1 «28» августа 2025 г.

Руководитель центра ИИ Коваленко А.В.

фамилия, инициалы



подпись

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

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

Сформировать у студентов систематизированные знания и практические навыки в области проектирования, разработки и оценки интеллектуальных агентов и мультиагентных систем, включая современные подходы на основе больших языковых моделей (LLM).

1.2 Задачи дисциплины

- Изучить архитектуры, модели взаимодействия, алгоритмы координации и кооперации агентов, а также математические основы мультиагентных алгоритмов.
- Освоить современные инструменты и фреймворки (OpenAI Agents SDK, LangChain, LangGraph, N8N, MCP) для создания работающих агентов и мультиагентных систем.
- Научиться применять мультиагентные подходы для решения реальных задач ИИ, включая интеграцию с внешними сервисами, управление памятью и состоянием, а также оценку эффективности.
- Получить опыт прототипирования и тестирования агентов в симулированных средах (Robocode, OpenAI Gymnasium) и анализа их коллективного поведения.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Мультиагентные системы» относится к дисциплинам, формируемым участниками образовательных отношений, код Б1.В.01.

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

- Промпт-инжиниринг в профессиональной деятельности;
- Обработка естественного языка.

Требованием к «входным» знаниям является понимание основ программирования на Python, Java.

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:

- | | |
|-------------|---|
| УК-1 | <i>Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач</i> |
| УК-1.1 | <p>Осуществляет поиск необходимой информации, опираясь на результаты анализа поставленной задачи</p> <p>Знать последние научные публикации и тренды в области мультиагентных систем и LLM.</p> <p>Уметь критически анализировать и синтезировать информацию из разных источников для создания инновационных решений.</p> <p>Владеть навыками работы с научной литературой (arXiv) и анализа state-of-the-art решений.</p> |
| УК-1.2 | <p>Выбирает оптимальный вариант решения задачи, аргументируя свой выбор</p> <p>Знать последние научные публикации и тренды в области мультиагентных систем и LLM.</p> <p>Уметь критически анализировать и синтезировать информацию из разных источников для создания инновационных решений.</p> <p>Владеть навыками работы с научной литературой (arXiv) и анализа state-of-the-art решений.</p> |
| УК-2 | <i>Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений</i> |
| УК-2.3 | <p>Использует принципы проектной методологии для решения профессиональных задач</p> <p>Знать передовые проектные методологии (Agile, Scrum) применительно к ИИ-разработке.</p> <p>Уметь управлять проектом от идеи до реализации, предвидеть и минимизировать риски.</p> <p>Владеть практиками Agile/Scrum в управлении ИИ-проектами и инструментами для управления рисками.</p> |
| УК-2.4 | <p>Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария</p> <p>Знать передовые проектные методологии (Agile, Scrum) применительно к ИИ-разработке.</p> <p>Уметь управлять проектом от идеи до реализации, предвидеть и минимизировать риски.</p> |

- Владеть практиками Agile/Scrum в управлении ИИ-проектами и инструментами для управления рисками.
- УК-4** *Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)*
- УК-4.3 Выбирает коммуникативно приемлемые стиль и средства взаимодействия в общении с деловыми партнерами
Знать психологические аспекты командной динамики и лидерства.
Уметь выступать лидером команды, распределять роли, мотивировать участников и разрешать конфликты.
- УК-4.4 Владеть навыками фасилитации, модерации и управления командой.
Ведет деловую переписку и использует диалог для сотрудничества в социальной и профессиональной сферах
Знать психологические аспекты командной динамики и лидерства.
Уметь выступать лидером команды, распределять роли, мотивировать участников и разрешать конфликты.
Владеть навыками фасилитации, модерации и управления командой.
- УК-6** *Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни*
- УК-6.1 Понимает необходимость осознанного управления своим временем и другими личностными ресурсами для выстраивания и реализации траектории саморазвития, личностных достижений, постоянного самообразования
Знать собственные сильные и слабые стороны в контексте изучаемой дисциплины.
Уметь осознанно выбирать направления для углубленного изучения в рамках курса.
Владеть методами саморефлексии и составления личного плана развития.
- ПК-1** *Способен активно участвовать в исследовании новых математических моделей в естественных науках; выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем*
- ПК-1.1 Разрабатывает концепцию и архитектуру программной системы, ее функциональные возможности и логику работы, делает выбор средств проектирования и реализации на основе требований с учетом существующих ограничений
Уметь проектировать архитектуры гибридных мультиагентных систем, интегрирующих LLM с классическими алгоритмами ИИ.
- ПК-1.2 Способен использовать знания о базовых принципах организации и основных этапах проектирования ИС
Уметь проектировать архитектуры гибридных мультиагентных систем, интегрирующих LLM с классическими алгоритмами ИИ.
- ПК-1.3 Использует методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения
Уметь проектировать архитектуры гибридных мультиагентных систем, интегрирующих LLM с классическими алгоритмами ИИ.

- ПК-4** *Способен планировать необходимые ресурсы и этапы выполнения работ в области информационно-коммуникационных технологий, составлять соответствующие технические описания и инструкции*
- ПК-4.1 Использует современные инструментальные средства разработки баз данных, прикладного программного обеспечения и систем различного функционального назначения
Знать внутреннее устройство и механизмы расширяемости используемых фреймворков.
Уметь создавать собственные инструменты и расширения для существующих фреймворков и оптимизировать производительность.
Владеть созданием кастомных MCP-серверов и углубленным программированием на LangGraph/LangChain.
- ПК-4.2 Применяет современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов на базе языков программирования, баз данных и пакетов прикладных программ
Знать внутреннее устройство и механизмы расширяемости используемых фреймворков.
Уметь создавать собственные инструменты и расширения для существующих фреймворков и оптимизировать производительность.
Владеть созданием кастомных MCP-серверов и углубленным программированием на LangGraph/LangChain.
- ПК-4.3 Способен использовать методы эффективного управления командой при разработке, внедрении и сопровождении программных продуктов
Знать внутреннее устройство и механизмы расширяемости используемых фреймворков.
Уметь создавать собственные инструменты и расширения для существующих фреймворков и оптимизировать производительность.
Владеть созданием кастомных MCP-серверов и углубленным программированием на LangGraph/LangChain.
- LLM-4** *Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей*
- LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов
Использует простейших агентов в пайплайнах.
Настраивает агентов и управляет их контекстом и задачами
- LLM-4.2 Интегрирует агентов с внешними сервисами
Подключает внешние функции и данные через wrapper.
Организует взаимодействие между агентом и внешними источниками
- LLM-4.3 Разрабатывает агентные паттерны
Использует паттерны «задай-выполни».
Реализует рассуждение на основе цепочек (ReAct, Plan&Solve)
- LLM-4.4 Управляет состоянием и памятью агентов
Использует кратковременную память и system prompts.
Настраивает и переключает долгосрочную/контекстную память
- LLM-4.5 Оценивает и оптимизирует эффективность агентов
Тестирует работу агента на стандартных сценариях.
Оценивает отклонения, настраивает поведение и порог доверия
- О-2** *Способен применять и (или) разрабатывать мультиагентные алгоритмы*
- О-2.3 Создает обученные интеллектуальные агенты, способные решать частные задачи ИИ и координировать свою работу с другими агентами

Выполняет обучение интеллектуальных агентов на основе имеющихся данных

FC-3 *Способен проводить фронтирные исследования в области управления, решения, агентных и мультиагентных систем*

FC-3.3 Исследует и создает мультиагентные системы
Реализует готовые VLA-модели (RT-2, GATO) для решения стандартных задач на стыке зрения, языка и действий. Применяет техники prompt engineering для адаптации моделей к конкретным сценариям. Интегрирует предобученные эмбединги текста и изображений в pipeline принятия решений агента.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 4 зач. ед. (144 часа), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)				
		7				
Контактная работа, в том числе:	72,3	72,3				
Аудиторные занятия (всего):	68	68				
Занятия лекционного типа	34	34				
Лабораторные занятия	34	34				
Занятия семинарского типа (семинары, практические занятия)						
Иная контактная работа:	2,3	2,3				
Контроль самостоятельной работы (КСР)	4	2				
Промежуточная аттестация (ИКР)	0,3	0,3				
Самостоятельная работа, в том числе:	36	36				
Проектная работа	33,4	33,4				
Выполнение индивидуальных заданий						
Реферат						
Подготовка к текущему контролю	2,3	2,3				
Контроль:						
Подготовка к экзамену	35,7	35,7				
Общая трудоемкость	час.	144	144			
	в том числе контактная работа	72,3	72,3			
	зач. ед	4	4			

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 7 семестре

№	Наименование разделов (тем)	Всего	Количество часов			Внеаудиторная работа СРС
			Аудиторная работа			
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Агентные системы	50	16		16	18
2.	Мультиагентные системы	54	18		18	18

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
ИТОГО по разделам дисциплины		104	34		34	36
Контроль самостоятельной работы (КСР)		4				
Промежуточная аттестация (ИКР)		0,3				
Подготовка к текущему контролю		35,7				
Общая трудоемкость по дисциплине		144				

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.	Агентные системы	Введение в агентные системы. Что такое интеллектуальный агент? Свойства агентов (автономность, реактивность, проактивность, социальность). Классификация агентов. LLM как когнитивный двигатель агента. Обзор предметной области и ключевых применений.	ЛР
2.	Агентные системы	Архитектура одиночного агента. Модель "Восприятие-Мышление-Действие". Агентные паттерны: "Задай-выполни" (Ask-and-Execute). Введение в reasoning: Chain-of-Thought (CoT). Понятие контекста и памяти.	ЛР
3.	Агентные системы	Продвинутые техники рассуждения для агентов. ReAct (Reasoning + Acting). Plan&Solve. Деревья размышлений (Tree of Thoughts). Сравнение эффективности разных методов.	ЛР
4.	Агентные системы	Управление памятью и состоянием агента. Кратковременная память (системные промпты, диалоговый буфер). Долговременная память (векторные базы	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		данных, RAG). Стратегии управления контекстом (суммаризация, фильтрация). LangChain, LangGraph и LangFlow.	
5.	Агентные системы	Инструменты и интеграции. Модель Compute Server (MCP) – концепция и преимущества. "Wrapper" как способ подключения внешних функций и данных. Безопасность и управление доступом агентов.	ЛР
6.	Агентные системы	Симуляции и воплощенные агенты. Роль симуляторов (PyBullet, MuJoCo, NVIDIA Isaac, Unity ML-Agents, OpenAI Gymnasium, Robocode). Обучение движению.	ЛР
7.	Агентные системы	Мультимодальные и VLA-агенты. Модели, объединяющие зрение, язык и действия (RT-2, GATO). Интеграция сенсорных модальностей в pipeline. Prompt engineering для мультимодальных сценариев.	ЛР
8.	Мультиагентные системы	Введение в мультиагентные системы (MAS). Определения и основные концепции. Преимущества и сложности. Типы взаимодействия: кооперация, координация, конкуренция, переговоры. Классические протоколы: FIPA, Contract Net.	ЛР
9.	Мультиагентные системы	Архитектуры мультиагентных систем. Плоские (децентрализованные) иерархические архитектуры. Фреймворки для MAS: JADE (Java), MASON. Подход OpenAI к мультиагентности.	ЛР
10.	Мультиагентные системы	Алгоритмы координации и кооперации. Проблема достижения консенсуса. Распределенное планирование задач. Методы группового принятия решений (голосование, взвешенное мнение). Задача византийских генералов.	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
11.	Мультиагентные системы	Роевой интеллект (Swarm Intelligence). Философия роевого поведения. Алгоритм роя частиц (PSO) – математические основы. Муравьиные алгоритмы (ACO) и пчелиные алгоритмы.	ЛР
12.	Мультиагентные системы	Применение роевых алгоритмов в ИИ. Оптимизация нейронных сетей. Кластеризация данных. Маршрутизация в сетях. Сравнение с традиционными методами (градиентный спуск, k-means).	ЛР
13.	Мультиагентные системы	Мультиагентное обучение с подкреплением (MARL). Основы RL: среда, агент, политика, функция ценности. Проблемы MARL: нестационарность, кредитное присвоение. Подходы: независимое Q-обучение, централизованное обучение с децентрализованным исполнением.	ЛР
14.	Мультиагентные системы	Коллективное поведение и фазовые переходы. Моделирование сложных систем. Параметры порядка и критические точки. Примеры: модель Изинга, модель Вика-Уилсона (мнение толпы).	ЛР
15.	Мультиагентные системы	Huawei Cangjie. Обзор архитектуры и возможностей. Сравнение с OpenAI Agents и LangChain. Кейсы применения в облачных и edge-системах.	ЛР
16.	Мультиагентные системы	Оценка эффективности агентных систем. Метрики для одиночных агентов (точность, задержка, стоимость). Метрики для MAS (пропускная способность системы, устойчивость). A/B тестирование и бенчмаркинг.	ЛР
17.	Мультиагентные системы	Подготовка к экзамену. Обзор идей для проектов. Разбор успешных кейсов. Принципы формирования команд и	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		планирования проекта. Ответы на вопросы.	

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.	Агентные системы	Настройка окружения (Python, Jupyter, виртуальные окружения). Первое знакомство с OpenAI Agent SDK. Создание простейшего агента.	ЛР
2.	Агентные системы	Паттерн "Задай-выполни" на практике. Работа с OpenAI Agents SDK. Создание агента с жестко заданным prompt. Управление контекстом разговора.	ЛР
3.	Агентные системы	Реализация ReAct-агента. Использование LangChain для построения цепочек. Интеграция простых инструментов (tool) в агента (например, калькулятор, поиск в интернете).	ЛР
4.	Агентные системы	Интеграция памяти в агента. Работа с векторными БД (ChromaDB/FAISS) через LangChain. Создание агента с RAG для ответов на вопросы по документам. LangGraph и LangFlow.	ЛР
5.	Агентные системы	Подключение внешних сервисов. Использование MCP-серверов. Создание wrapper'ов для REST API (например, погода, биржевые котировки). Знакомство с N8N для оркестрации workflow.	ЛР
6.	Агентные системы	Знакомство с Robocode. Создание простого танка на Java. Программирование реактивного	ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
		поведения (избегание стен, преследование цели). Введение в OpenAI Gymnasium. Обучение одиночного агента в среде Robocode. Знакомство со Stable Baselines3.	
7.	Агентные системы	Работа с мультимодальными моделями. Использование OpenAI GPT или аналогов для анализа изображений. Создание агента, который описывает изображение и отвечает на вопросы о нем.	ЛР
8.	Мультиагентные системы	Первые шаги в мультиагентности. Создание двух простых агентов на LangGraph. Реализация простейшего сценария координации (например, запрос-подтверждение).	ЛР
9.	Мультиагентные системы	Создание мультиагентного чата. Моделирование дискуссии между агентами с разными ролями (аналитик, критик, генератор идей) с использованием LangGraph.	ЛР
10.	Мультиагентные системы	Реализация протокола Contract Net. Программирование на Java в среде JADE. Создание системы, где один агент объявляет задачу, а другие подают заявки на ее выполнение.	ЛР
11.	Мультиагентные системы	Решение задачи оптимизации с помощью PSO. Реализация PSO на Python для поиска минимума функции. Визуализация движения роя.	ЛР
12.	Мультиагентные системы	Муравьиный алгоритм для задачи коммивояжера. Реализация ACO на Python. Сравнение эффективности с жадными алгоритмами.	ЛР
13.	Мультиагентные системы	Моделирование коллективного поведения. Создание симуляции по модели "Согласование мнений".	ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
14.	Мультиагентные системы	Экосистема Huawei Cangjie. Изучение документации и API. Создание простого агента для работы с данными на платформе Cangjie.	ЛР
15.	Мультиагентные системы	Тестирование и отладка агентов. Создание набора стандартных сценариев (eval-набор). Анализ логов агентов, вычисление метрик. Настройка "порога доверия".	ЛР
16.	Мультиагентные системы	Работа над командным проектом. Формирование команд, выбор темы. Разработка технического задания и архитектуры проекта. Консультации.	ЛР
17.	Мультиагентные системы	Работа над командным проектом.	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. В качестве курсового проекта студенты разрабатывают мультиагентную систему по заданию от индустриального партнера.

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
7	ЛР	Занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	12
Итого			12

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 «Мультиагентные системы».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме оценки лабораторных работ и 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	Агентные системы	УК-1.1; УК-1.2; УК-2.3; УК-2.4; УК-4.3; УК-4.4; УК-6.1; ПК-1.1; ПК-1.2; ПК-1.3; ПК-4.1; ПК-4.2; ПК-4.3;	ЛР 1-7	Проект с кейсом от индустриального партнера
2	Мультиагентные	LLM-4.1; LLM-4.2; LLM-4.3; LLM-4.4; LLM-4.5; О-2.3; FC-3.3	ЛР 8-14	Проект с кейсом от индустриального партнера

Показатели, критерии и шкала оценки сформированных компетенций

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

- УК-1** *Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач*
- УК-1.1 Осуществляет поиск необходимой информации, опираясь на результаты анализа поставленной задачи
Знать основные источники информации по агентным системам (документация, учебники).
Уметь находить готовые примеры кода и адаптировать их для решения типовых задач.
Владеть поиском в интернете и документации; базовым чтением и модификацией кода на Python.
- УК-1.2 Выбирает оптимальный вариант решения задачи, аргументируя свой выбор
Знать основные источники информации по агентным системам (документация, учебники).
Уметь находить готовые примеры кода и адаптировать их для решения типовых задач.
Владеть поиском в интернете и документации; базовым чтением и модификацией кода на Python.
- УК-2** *Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений*
- УК-2.3 Использует принципы проектной методологии для решения профессиональных задач
Знать основные этапы проектной деятельности и простые критерии выбора инструментов.
Уметь следовать готовому техническому заданию проекта и выбирать известный инструмент из предложенного списка.
Владеть методологией выполнения задания по инструкции.
- УК-2.4 Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария
Знать основные этапы проектной деятельности и простые критерии выбора инструментов.
Уметь следовать готовому техническому заданию проекта и выбирать известный инструмент из предложенного списка.
Владеть методологией выполнения задания по инструкции.

УК-4	<i>Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)</i>
УК-4.3	Выбирает коммуникативно приемлемые стиль и средства взаимодействия в общении с деловыми партнерами Знать основные правила коммуникации в команде. Уметь участвовать в обсуждении проекта в чате и представлять свою часть работы. Владеть навыками письменной коммуникации в мессенджерах и оформления простой документации.
УК-4.4	Ведет деловую переписку и использует диалог для сотрудничества в социальной и профессиональной сферах Знать основные правила коммуникации в команде. Уметь участвовать в обсуждении проекта в чате и представлять свою часть работы. Владеть навыками письменной коммуникации в мессенджерах и оформления простой документации.
УК-6	<i>Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни</i>
УК-6.1	Понимает необходимость осознанного управления своим временем и другими личностными ресурсами для выстраивания и реализации траектории саморазвития, личностных достижений, постоянного самообразования Знать сроки и основные этапы проектной работы по предмету. Уметь планировать личное время для выполнения этапов проекта в установленные сроки. Владеть методами тайм-менеджмента (например, составление простого плана работ).
ПК-1	<i>Способен активно участвовать в исследовании новых математических моделей в естественных науках; выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем</i>
ПК-1.1	Разрабатывает концепцию и архитектуру программной системы, ее функциональные возможности и логику работы, делает выбор средств проектирования и реализации на основе требований с учетом существующих ограничений Знать базовые архитектуры агентов (ReAct) и мультиагентных систем (централизованная, децентрализованная). Уметь описывать архитектуру своего проекта, используя стандартные блоки, и применять готовые методологии для простых задач. Владеть LangChain для создания цепочек и OpenAI API для вызова моделей.
ПК-1.2	Способен использовать знания о базовых принципах организации и основных этапах проектирования ИС Знать базовые архитектуры агентов (ReAct) и мультиагентных систем (централизованная, децентрализованная). Уметь описывать архитектуру своего проекта, используя стандартные блоки, и применять готовые методологии для простых задач. Владеть LangChain для создания цепочек и OpenAI API для вызова моделей.
ПК-1.3	Использует методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и

- использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения
Знать базовые архитектуры агентов (ReAct) и мультиагентных систем (централизованная, децентрализованная).
Уметь описывать архитектуру своего проекта, используя стандартные блоки, и применять готовые методологии для простых задач.
Владеть LangChain для создания цепочек и OpenAI API для вызова моделей.
- ПК-4** *Способен планировать необходимые ресурсы и этапы выполнения работ в области информационно-коммуникационных технологий, составлять соответствующие технические описания и инструкции*
- ПК-4.1 Использует современные инструментальные средства разработки баз данных, прикладного программного обеспечения и систем различного функционального назначения
Знать назначение основных инструментов курса (OpenAI SDK, LangChain).
Уметь использовать изученные инструменты по предоставленным шаблонам.
Владеть OpenAI Agents SDK и базовыми возможностями LangChain.
- ПК-4.2 Применяет современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов на базе языков программирования, баз данных и пакетов прикладных программ
Знать назначение основных инструментов курса (OpenAI SDK, LangChain).
Уметь использовать изученные инструменты по предоставленным шаблонам.
Владеть OpenAI Agents SDK и базовыми возможностями LangChain.
- ПК-4.3 Способен использовать методы эффективного управления командой при разработке, внедрении и сопровождении программных продуктов
Знать назначение основных инструментов курса (OpenAI SDK, LangChain).
Уметь использовать изученные инструменты по предоставленным шаблонам.
Владеть OpenAI Agents SDK и базовыми возможностями LangChain.
- LLM-4** *Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей*
- LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов
Использует простейших агентов в пайплайнах.
Настраивает агентов и управляет их контекстом и задачами
- LLM-4.2 Интегрирует агентов с внешними сервисами
Подключает внешние функции и данные через wrapper.
Организует взаимодействие между агентом и внешними источниками
- LLM-4.3 Разрабатывает агентные паттерны
Использует паттерны «задай-выполни».
Реализует рассуждение на основе цепочек (ReAct, Plan&Solve)
- LLM-4.4 Управляет состоянием и памятью агентов
Использует кратковременную память и system prompts.
Настраивает и переключает долгосрочную/контекстную память
- LLM-4.5 Оценивает и оптимизирует эффективность агентов
Тестирует работу агента на стандартных сценариях.
Оценивает отклонения, настраивает поведение и порог доверия
- О-2** *Способен применять и (или) разрабатывать мультиагентные алгоритмы*
- О-2.3 Создает обученные интеллектуальные агенты, способные решать частные задачи ИИ и координировать свою работу с другими агентами
Выполняет обучение интеллектуальных агентов на основе имеющихся данных
- ФС-3** *Способен проводить фронтальные исследования в области управления, решения, агентных и мультиагентных систем*
- ФС-3.3 Исследует и создает мультиагентные системы

Реализует готовые VLA-модели (RT-2, GATO) для решения стандартных задач на стыке зрения, языка и действий. Применяет техники prompt engineering для адаптации моделей к конкретным сценариям. Интегрирует предобученные эмбединги текста и изображений в pipeline принятия решений агента.

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

- УК-1** *Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач*
- УК-1.1 Осуществляет поиск необходимой информации, опираясь на результаты анализа поставленной задачи
Знать плюсы и минусы разных подходов и инструментов для построения агентов.
Уметь самостоятельно находить и сравнивать несколько решений для поставленной задачи, аргументировать выбор.
Владеть анализом GitHub репозитория, технических статей и документации.
- УК-1.2 Выбирает оптимальный вариант решения задачи, аргументируя свой выбор
Знать плюсы и минусы разных подходов и инструментов для построения агентов.
Уметь самостоятельно находить и сравнивать несколько решений для поставленной задачи, аргументировать выбор.
Владеть анализом GitHub репозитория, технических статей и документации.
- УК-2** *Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений*
- УК-2.3 Использует принципы проектной методологии для решения профессиональных задач
Знать принципы оценки ресурсов (время, стоимость API) и рисков (отказ агента) для проекта.
Уметь самостоятельно декомпозировать задачу на подзадачи и оценивать необходимые ресурсы.
Владеть методами декомпозиции задач и базового проектного планирования.
- УК-2.4 Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария
Знать принципы оценки ресурсов (время, стоимость API) и рисков (отказ агента) для проекта.
Уметь самостоятельно декомпозировать задачу на подзадачи и оценивать необходимые ресурсы.
Владеть методами декомпозиции задач и базового проектного планирования.
- УК-4** *Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)*
- УК-4.3 Выбирает коммуникативно приемлемые стиль и средства взаимодействия в общении с деловыми партнерами
Знать эффективные практики командной работы (code review, stand-up).
Уметь координировать свои задачи с другими членами команды и вести техническую дискуссию.
Владеть инструментами коллаборации (Git, сервис-деск) и навыками ведения технических обсуждений.
- УК-4.4 Ведет деловую переписку и использует диалог для сотрудничества в социальной и профессиональной сферах

	<p>Знать эффективные практики командной работы (code review, stand-up). Уметь координировать свои задачи с другими членами команды и вести техническую дискуссию. Владеть инструментами коллаборации (Git, сервис-деск) и навыками ведения технических обсуждений.</p>
УК-6	<i>Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни</i>
УК-6.1	<p>Понимает необходимость осознанного управления своим временем и другими личностными ресурсами для выстраивания и реализации траектории саморазвития, личностных достижений, постоянного самообразования Знать собственные сильные и слабые стороны в контексте изучаемой дисциплины. Уметь осознанно выбирать направления для углубленного изучения в рамках курса.</p>
ПК-1	<i>Способен активно участвовать в исследовании новых математических моделей в естественных науках; выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем</i>
ПК-1.1	<p>Разрабатывает концепцию и архитектуру программной системы, ее функциональные возможности и логику работы, делает выбор средств проектирования и реализации на основе требований с учетом существующих ограничений Знать достоинства и недостатки различных архитектур (иерархическая, плоская, федеративная). Уметь проектировать архитектуру мультиагентной системы для типовой задачи и выбирать подходящие протоколы взаимодействия. Владеть языком проектирования UML (диаграммы последовательностей, компонентов) и фреймворками JADE или LangGraph.</p>
ПК-1.2	<p>Способен использовать знания о базовых принципах организации и основных этапах проектирования ИС Знать достоинства и недостатки различных архитектур (иерархическая, плоская, федеративная). Уметь проектировать архитектуру мультиагентной системы для типовой задачи и выбирать подходящие протоколы взаимодействия. Владеть языком проектирования UML (диаграммы последовательностей, компонентов) и фреймворками JADE или LangGraph.</p>
ПК-1.3	<p>Использует методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения Знать достоинства и недостатки различных архитектур (иерархическая, плоская, федеративная). Уметь проектировать архитектуру мультиагентной системы для типовой задачи и выбирать подходящие протоколы взаимодействия. Владеть языком проектирования UML (диаграммы последовательностей, компонентов) и фреймворками JADE или LangGraph.</p>
ПК-4	<i>Способен планировать необходимые ресурсы и этапы выполнения работ в области информационно-коммуникационных технологий, составлять соответствующие технические описания и инструкции</i>

- ПК-4.1 Использует современные инструментальные средства разработки баз данных, прикладного программного обеспечения и систем различного функционального назначения
Знать глубоко возможности и ограничения инструментов (LangChain, LangGraph, N8N).
Уметь комбинировать несколько инструментов для создания сложного пайплайна и интегрировать MCP-серверы.
Владеть LangGraph для создания циклических workflow, N8N для оркестрации и моделью MCP (Model Context Protocol).
- ПК-4.2 Применяет современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов на базе языков программирования, баз данных и пакетов прикладных программ
Знать глубоко возможности и ограничения инструментов (LangChain, LangGraph, N8N).
Уметь комбинировать несколько инструментов для создания сложного пайплайна и интегрировать MCP-серверы.
Владеть LangGraph для создания циклических workflow, N8N для оркестрации и моделью MCP (Model Context Protocol).
- ПК-4.3 Способен использовать методы эффективного управления командой при разработке, внедрении и сопровождении программных продуктов
Знать глубоко возможности и ограничения инструментов (LangChain, LangGraph, N8N).
Уметь комбинировать несколько инструментов для создания сложного пайплайна и интегрировать MCP-серверы.
Владеть LangGraph для создания циклических workflow, N8N для оркестрации и моделью MCP (Model Context Protocol).
- LLM-4 Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей**
- LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов
Использует простейших агентов в пайплайнах.
Настраивает агентов и управляет их контекстом и задачами
- LLM-4.2 Интегрирует агентов с внешними сервисами
Подключает внешние функции и данные через wrapper.
Организует взаимодействие между агентом и внешними источниками
- LLM-4.3 Разрабатывает агентные паттерны
Использует паттерны «задай-выполни».
Реализует рассуждение на основе цепочек (ReAct, Plan&Solve)
- LLM-4.4 Управляет состоянием и памятью агентов
Использует кратковременную память и system prompts.
Настраивает и переключает долгосрочную/контекстную память
- LLM-4.5 Оценивает и оптимизирует эффективность агентов
Тестирует работу агента на стандартных сценариях.
Оценивает отклонения, настраивает поведение и порог доверия
- O-2 Способен применять и (или) разрабатывать мультиагентные алгоритмы**
- O-2.3 Создает обученные интеллектуальные агенты, способные решать частные задачи ИИ и координировать свою работу с другими агентами
Выполняет обучение интеллектуальных агентов на основе имеющихся данных
- FC-3 Способен проводить фронтальные исследования в области управления, решения, агентных и мультиагентных систем**
- FC-3.3 Исследует и создает мультиагентные системы

Реализует готовые VLA-модели (RT-2, GATO) для решения стандартных задач на стыке зрения, языка и действий. Применяет техники prompt engineering для адаптации моделей к конкретным сценариям. Интегрирует предобученные эмбединги текста и изображений в pipeline принятия решений агента.

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

- УК-1** *Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач*
- УК-1.1 Осуществляет поиск необходимой информации, опираясь на результаты анализа поставленной задачи
Знать последние научные публикации и тренды в области мультиагентных систем и LLM.
Уметь критически анализировать и синтезировать информацию из разных источников для создания инновационных решений.
Владеть навыками работы с научной литературой (arXiv) и анализа state-of-the-art решений.
- УК-1.2 Выбирает оптимальный вариант решения задачи, аргументируя свой выбор
Знать последние научные публикации и тренды в области мультиагентных систем и LLM.
Уметь критически анализировать и синтезировать информацию из разных источников для создания инновационных решений.
Владеть навыками работы с научной литературой (arXiv) и анализа state-of-the-art решений.
- УК-2** *Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений*
- УК-2.3 Использует принципы проектной методологии для решения профессиональных задач
Знать передовые проектные методологии (Agile, Scrum) применительно к ИИ-разработке.
Уметь управлять проектом от идеи до реализации, предвидеть и минимизировать риски.
Владеть практиками Agile/Scrum в управлении ИИ-проектами и инструментами для управления рисками.
- УК-2.4 Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария
Знать передовые проектные методологии (Agile, Scrum) применительно к ИИ-разработке.
Уметь управлять проектом от идеи до реализации, предвидеть и минимизировать риски.
Владеть практиками Agile/Scrum в управлении ИИ-проектами и инструментами для управления рисками.
- УК-4** *Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)*
- УК-4.3 Выбирает коммуникативно приемлемые стиль и средства взаимодействия в общении с деловыми партнерами
Знать психологические аспекты командной динамики и лидерства.
Уметь выступать лидером команды, распределять роли, мотивировать участников и разрешать конфликты.

- УК-4.4 Владеть навыками фасилитации, модерации и управления командой.
Ведет деловую переписку и использует диалог для сотрудничества в социальной и профессиональной сферах
Знать психологические аспекты командной динамики и лидерства.
Уметь выступать лидером команды, распределять роли, мотивировать участников и разрешать конфликты.
- УК-6 Владеть навыками фасилитации, модерации и управления командой.
Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
- УК-6.1 Понимает необходимость осознанного управления своим временем и другими личностными ресурсами для выстраивания и реализации траектории саморазвития, личностных достижений, постоянного самообразования
Знать собственные сильные и слабые стороны в контексте изучаемой дисциплины.
Уметь осознанно выбирать направления для углубленного изучения в рамках курса.
Владеть методами саморефлексии и составления личного плана развития.
- ПК-1 **Способен активно участвовать в исследовании новых математических моделей в естественных науках; выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем**
- ПК-1.1 Разрабатывает концепцию и архитектуру программной системы, ее функциональные возможности и логику работы, делает выбор средств проектирования и реализации на основе требований с учетом существующих ограничений
Уметь проектировать архитектуры гибридных мультиагентных систем, интегрирующих LLM с классическими алгоритмами ИИ.
- ПК-1.2 Способен использовать знания о базовых принципах организации и основных этапах проектирования ИС
Уметь проектировать архитектуры гибридных мультиагентных систем, интегрирующих LLM с классическими алгоритмами ИИ.
- ПК-1.3 Использует методологии разработки программного обеспечения и технологии программирования, методологии и технологии проектирования и использования баз данных, методы и средства проектирования программных интерфейсов, принципы построения архитектуры программного обеспечения
Уметь проектировать архитектуры гибридных мультиагентных систем, интегрирующих LLM с классическими алгоритмами ИИ.
- ПК-4 **Способен планировать необходимые ресурсы и этапы выполнения работ в области информационно-коммуникационных технологий, составлять соответствующие технические описания и инструкции**
- ПК-4.1 Использует современные инструментальные средства разработки баз данных, прикладного программного обеспечения и систем различного функционального назначения
Знать внутреннее устройство и механизмы расширяемости используемых фреймворков.
Уметь создавать собственные инструменты и расширения для существующих фреймворков и оптимизировать производительность.
Владеть созданием кастомных MCP-серверов и углубленным программированием на LangGraph/LangChain.

- ПК-4.2 Применяет современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов на базе языков программирования, баз данных и пакетов прикладных программ
Знать внутреннее устройство и механизмы расширяемости используемых фреймворков.
Уметь создавать собственные инструменты и расширения для существующих фреймворков и оптимизировать производительность.
Владеть созданием кастомных MCP-серверов и углубленным программированием на LangGraph/LangChain.
- ПК-4.3 Способен использовать методы эффективного управления командой при разработке, внедрении и сопровождении программных продуктов
Знать внутреннее устройство и механизмы расширяемости используемых фреймворков.
Уметь создавать собственные инструменты и расширения для существующих фреймворков и оптимизировать производительность.
Владеть созданием кастомных MCP-серверов и углубленным программированием на LangGraph/LangChain.
- LLM-4** ***Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей***
- LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов
Использует простейших агентов в пайплайнах.
Настраивает агентов и управляет их контекстом и задачами
- LLM-4.2 Интегрирует агентов с внешними сервисами
Подключает внешние функции и данные через wrapper.
Организует взаимодействие между агентом и внешними источниками
- LLM-4.3 Разрабатывает агентные паттерны
Использует паттерны «задай-выполни».
Реализует рассуждение на основе цепочек (ReAct, Plan&Solve)
- LLM-4.4 Управляет состоянием и памятью агентов
Использует кратковременную память и system prompts.
Настраивает и переключает долгосрочную/контекстную память
- LLM-4.5 Оценивает и оптимизирует эффективность агентов
Тестирует работу агента на стандартных сценариях.
Оценивает отклонения, настраивает поведение и порог доверия
- O-2** ***Способен применять и (или) разрабатывать мультиагентные алгоритмы***
- O-2.3 Создает обученные интеллектуальные агенты, способные решать частные задачи ИИ и координировать свою работу с другими агентами
Выполняет обучение интеллектуальных агентов на основе имеющихся данных
- FC-3** ***Способен проводить фронтальные исследования в области управления, решения, агентных и мультиагентных систем***
- FC-3.3 Исследует и создает мультиагентные системы
Реализует готовые VLA-модели (RT-2, GATO) для решения стандартных задач на стыке зрения, языка и действий. Применяет техники prompt engineering для адаптации моделей к конкретным сценариям. Интегрирует предобученные эмбединги текста и изображений в pipeline принятия решений агента.

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

1. Настройка окружения (Python, Jupyter, виртуальные окружения).
Первое знакомство с OpenAI API.
Создание простейшего чат-бота.
2. Паттерн "Задай-выполни" на практике.
Работа с OpenAI Agents SDK.
Создание агента с жестко заданным prompt.
Управление контекстом разговора.
Реализация ReAct-агента.
3. Использование LangChain для построения цепочек.
Интеграция простых инструментов (tool) в агента (например, калькулятор, поиск в интернете).
4. Интеграция памяти в агента.
Работа с векторными БД (ChromaDB/FAISS) через LangChain.
Создание агента с RAG для ответов на вопросы по документам. LangGraph и LangFlow.
5. Подключение внешних сервисов.
Использование MCP-серверов.
Создание wrapper'ов для REST API (например, погода, биржевые котировки).
Знакомство с N8N для оркестрации workflow.
6. Знакомство с Robocode.
Создание простого танка на Java.
Программирование реактивного поведения (избегание стен, преследование цели).
Введение в OpenAI Gymnasium.
Обучение одиночного агента в среде Robocode.
Знакомство со Stable Baselines3.
7. Работа с мультимодальными моделями.
Использование OpenAI GPT-4V или аналогов для анализа изображений.
Создание агента, который описывает изображение и отвечает на вопросы о нем.
8. Первые шаги в мультиагентности.
Создание двух простых агентов на LangGraph.
Реализация простейшего сценария координации (например, запрос-подтверждение).
Создание мультиагентного чата.
Моделирование дискуссии между агентами с разными ролями (аналитик, критик, генератор идей) с использованием LangGraph.
9. Реализация протокола Contract Net.
Программирование на Java в среде JADE.
Создание системы, где один агент объявляет задачу, а другие подают заявки на ее выполнение.
10. Решение задачи оптимизации с помощью PSO.
Реализация PSO на Python для поиска минимума функции.
11. Визуализация движения роя.

Муравьиный алгоритм для задачи коммивояжера.

12. Реализация АСО на Python.
Сравнение эффективности с жадными алгоритмами.

13. Моделирование коллективного поведения.
Создание симуляции по модели "Согласование мнений".

14. Экосистема Huawei Cangjie.
Изучение документации и API.
Создание простого агента для работы с данными на платформе Cangjie.

15. Тестирование и отладка агентов.
Создание набора стандартных сценариев (eval-набор).
Анализ логов агентов, вычисление метрик. Настройка "порога доверия".

16. Работа над командным проектом.
Формирование команд, выбор темы.
Разработка технического задания и архитектуры проекта. Консультации.

17. Работа над командным проектом.

Пример лабораторной работы

Лабораторная работа №1: Лабораторная работа №1: Создание базового агента с использованием OpenAI Agents SDK и YandexGPT

Цель работы

Освоить принципы создания интеллектуальных агентов с использованием OpenAI Agents SDK.

Задачи работы

1. Настроить рабочее окружение для работы с OpenAI Agents SDK
2. Создать базового агента с использованием локальной или облачной LLM
3. Протестировать агента

Ожидаемый результат

После выполнения лабораторной работы студент должен предоставить:

Кодовая база:

Python-скрипт с реализацией агента на OpenAI Agents SDK

Конфигурационные файлы

Файл зависимостей (requirements.txt)

Документация:

Технический отчет содержащий:

Описание агента

Примеры работы

Критерии оценки

Зачтено: Агент запускается, работает как показано в отчете.
Не зачтено: Агент не работает или отчет не предоставлен.

Рекомендуемые инструменты
- GIT, Python, Llama или YandexGPT.

Ход работы

Шаг 1: Установка Ollama

Сначала установите Ollama, следуя инструкциям для вашей операционной системы:
Для macOS и Linux:

```
curl -fsSL https://ollama.ai/install.sh | sh
```

Для Windows:

Скачайте установщик с сайта Ollama.

Шаг 2: Загрузка модели

Загрузите подходящую модель, которая будет работать с вашим агентом. Например Mistral:

```
ollama pull mistral
```

Проверьте, что Ollama работает, выполнив:

```
ollama run mistral "Hello, are you running correctly?"
```

Вы должны увидеть ответ, сгенерированный моделью.

Шаг 3: Установка OpenAI Agents SDK

Клонировать репозиторий и установить пакет:

```
git clone https://github.com/openai/openai-agents-python.git
cd openai-agents-python
pip install -e .
```

Это устанавливает пакет в режиме разработки, позволяя при необходимости изменять код.

Шаг 4: Установка необходимых зависимостей

Установите дополнительные зависимости:

```
pip install requests python-dotenv pydantic
```

Интеграция Ollama с OpenAI Agents SDK

OpenAI Agents SDK использует клиент OpenAI Python. Нам нужно создать кастомный клиент,

```

# который перенаправляет запросы к Оллма вместо серверов OpenAI.

# Шаг 1: Создание кастомного клиента

# Создайте файл ollama_client.py:

import os
from openai import OpenAI

class OllamaClient(OpenAI):
    """Кастомный клиент OpenAI, который перенаправляет запросы в Оллма."""

    def __init__(self, model_name="mistral", **kwargs):
        # Настройка для использования эндпоинта Оллма
        kwargs["base_url"] = "http://localhost:11434/v1"

        # Оллма не требует API-ключ, но клиент ожидает его наличие
        kwargs["api_key"] = "ollama-placeholder-key"

        super().__init__(**kwargs)
        self.model_name = model_name

        # Проверка существования модели
        print(f"Используется модель Оллма: {model_name}")

    def create_completion(self, *args, **kwargs):
        # Переопределяем имя модели, если оно не указано явно
        if "model" not in kwargs:
            kwargs["model"] = self.model_name

        return super().create_completion(*args, **kwargs)

    def create_chat_completion(self, *args, **kwargs):
        # Переопределяем имя модели, если оно не указано явно
        if "model" not in kwargs:
            kwargs["model"] = self.model_name

        return super().create_chat_completion(*args, **kwargs)

# Эти методы нужны для совместимости с библиотекой agents
def completion(self, prompt, **kwargs):
    if "model" not in kwargs:
        kwargs["model"] = self.model_name
    return self.completions.create(prompt=prompt, **kwargs)

def chat_completion(self, messages, **kwargs):
    if "model" not in kwargs:
        kwargs["model"] = self.model_name
    return self.chat.completions.create(messages=messages, **kwargs)

# Шаг 2: Создание адаптера для OpenAI Agents SDK

```

```

# Теперь создадим адаптер для совместимости OpenAI Agents SDK с нашим Ollama
клиентом.
# Создайте файл agent_adapter.py:

from ollama_client import OllamaClient
from openai.types.chat import ChatCompletion, ChatCompletionMessage
import agents.agent as agent_module
from agents.agent import Agent
from agents.run import Runner, RunConfig
from agents.models import _openai_shared
import json
import logging

# Настройка логирования
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(name)s - %(levelname)s -
%(message)s')
logger = logging.getLogger(__name__)

# Устанавливаем заглушку для OpenAI API ключа чтобы избежать ошибок инициализации
_openai_shared.set_default_openai_key("placeholder-key")

# Сохраняем оригинальный init для класса Agent
original_init = Agent.__init__

def patched_init(self, *args, **kwargs):
    """Заменяем модель на OllamaClient если не предоставлена."""
    if "model" not in kwargs:
        kwargs["model"] = OllamaClient(model_name="mistral")
    original_init(self, *args, **kwargs)

# Применяем патч к init
Agent.__init__ = patched_init

# Класс для структурированного вызова инструментов
class ToolCall:
    def __init__(self, name, inputs=None):
        self.name = name
        self.inputs = inputs or {}

# Определяем класс ответа, соответствующий ожиданиям main.py
class AgentResponse:
    def __init__(self, result):
        # Извлекаем сообщение из финального вывода
        if hasattr(result, 'final_output'):
            if isinstance(result.final_output, str):
                self.message = result.final_output
            else:
                self.message = str(result.final_output)
        else:
            self.message = "Извините, я не смог обработать этот запрос."

```

```

# Получаем ID диалога если доступен
self.conversation_id = getattr(result, 'conversation_id', None)

# Инициализируем tool_calls
self.tool_calls = []

# Извлекаем вызовы инструментов из raw_responses
if hasattr(result, 'raw_responses'):
    for response in result.raw_responses:
        try:
            if hasattr(response, 'output') and hasattr(response.output, 'tool_calls'):
                for tool_call in response.output.tool_calls:
                    # Обрабатываем случай, когда tool_call является словарем
                    if isinstance(tool_call, dict):
                        name = tool_call.get('name', 'unknown_tool')
                        inputs = tool_call.get('inputs', {})
                        self.tool_calls.append(ToolCall(name, inputs))
                    else:
                        # Предполагаем, что это уже объект с атрибутами name и inputs
                        self.tool_calls.append(tool_call)
        except Exception as e:
            logger.error(f"Ошибка извлечения вызовов инструментов: {str(e)}")

# Добавляем метод run в класс Agent
def run(self, message, conversation_id=None):
    """Запуск агента с заданным сообщением.

    Args:
        message: Пользовательское сообщение для обработки
        conversation_id: Опциональный ID диалога для продолжения беседы

    Returns:
        Объект ответа с атрибутами message, conversation_id и tool_calls
    """
    try:
        # Создаем прямой промпт для модели
        prompt = f"""
        {self.instructions}

        Пользовательский запрос: {message}"""

```

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

Экзамен представляет собой защиту командного проекта, который должен продемонстрировать все ключевые компетенции, полученные в ходе курса.

Темы проектов выбираются в соответствии с заданиями промышленных партнеров и могут быть реализованы, например, в следующем виде:

Мультиагентная система анализа отзывов: Агенты-аналитики, агент-суммаризатор, агент-классификатор, работающие через LangGraph.

Команда ИИ-разработчиков: Система из нескольких LLM-агентов (архитектор, кодер, тестировщик), которая по ТЗ создает прототип веб-приложения.

Агент-консьерж: Интеграция через N8N и MCP нескольких сервисов (календарь, почта, доставка еды, погода) в единого персонального ассистента.

Критерии оценки:

Корректность применения инструментов и алгоритмов, рабочий код (1 балл).

Умение интегрировать различные компоненты и сервисы (1 балл).

Презентация и защита проекта (1 балл).

Сложность и проработанность архитектуры (1 балл).

Качество кода и документации (1 балл).

Результат: Команда представляет работающий прототип, документацию по архитектуре, репозиторий с кодом, а также отчет по результатам работы системы

Перечень компетенций (части компетенции), проверяемых оценочным средством
УК-1.1; УК-1.2; УК-2.3; УК-2.4; УК-4.3; УК-4.4; УК-6.1; ПК-1.1; ПК-1.2; ПК-1.3; ПК-4.1; ПК-4.2; ПК-4.3; LLM-4.1; LLM-4.2; LLM-4.3; LLM-4.4; LLM-4.5; О-2.3; FC-3.3

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 (защита проекта).

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:

Удовлетворительно – выполнено 60% лабораторных работ и выполнен проект на 3 или более баллов.

Хорошо – выполнено 75% лабораторных работ и выполнен проект мультиагентной системы на 4 или более баллов.

Отлично – выполнено 90% лабораторных работ и выполнен проект мультиагентной системы на 5 баллов.

Не удовлетворительно – 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 4-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab, Google Colab или Yandex DataSphere, JupyterHub, Hugging Face).
- Разработаны лабораторные работы;
- Инфраструктура для приёма задач (gitlab, CI/CD) согласована с лабораторными работами.

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

- дать представление о работе в IT инфраструктуре (приучить пользоваться гитом, jupyter-ноутбуками).

Задачи преподавателя:

- Организация регистрации студентов в Google Colab и Yandex DataSphere
- Создание учетных записей студентов в gitlab вуза;
- Настройка GitLab Runner для автоматического тестирования кода.
- Разработка шаблонного репозитория для лабораторных работ с предустановленными зависимостями (PyTorch, Keras, TensorFlow, Hugging Face Transformers).
- Написание автотестов для проверки корректности выполнения заданий (например, fine-tuning моделей).
- Визуализация результатов тестирования через HTML-отчеты.
- Подготовка инструкций по работе с Git и облачными ресурсами.

Ожидаемые результаты студентов:

- начальное представление о работе в IT инфраструктуре (гит, нейминг).
- Навыки запуска и тестирования глубоких нейросетей в облачных средах.
- Понимание CI/CD-процессов в контексте разработки глубоких нейросетей.

Порядок реализации

Задача №1: Организация регистрации студентов в Google Colab и Yandex DataSphere

Задача №2: Создание учетных записей студентов в gitlab вуза

Задача №3: Настройка GitLab Runner:

Для автоматического тестирования кода используется Docker-образ с предустановленными библиотеками (PyTorch, Keras, TensorFlow, Hugging Face Transformers).

Для выполнения CI/CD пайплайна был настроен GitLab Runner на удаленной 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.

Задача №4: Шаблонный репозиторий:

Включает:

.gitlab-ci.yml для CI/CD.

Скрипты для предобработки текста и обучения моделей.

Примеры кода для работы с BERT, GPT и другими архитектурами.

Задача №5: Автотесты:

Проверяют корректность fine-tuning моделей (например, ассурасу на тестовом датасете).

Задача №6: Визуализация результатов:

Генерация HTML-отчетов с результатами тестирования, включая метрики качества моделей.

Порядок проверки корректности:

- Наличие Git-репозитория у всех студентов.
- Шаблонный репозиторий с подключенными автотестами.
- Инструкция по работе с Git и CI/CD в формате README.md.

Вся структура максимально адаптирована для копирования студентами и минимизации порога входа при выполнении лабораторных

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Курс рассчитан на студентов 4-го года обучения.

- Наличие доступа к вычислительным ресурсам (GitLab) и к GPU/CPU (Kaggle, локальные серверы).
- Разработана инфраструктура для приёма задач (Gitlab, CI/CD) и согласована с лабораторными работами и настроена на всех студентов образовательной программы;

Цели, задачи и ожидаемые результаты

Цели организации лабораторных работ:

- Изучить архитектуры, модели взаимодействия, алгоритмы координации и кооперации агентов, а также математические основы мультиагентных алгоритмов.
- Освоить современные инструменты и фреймворки (OpenAI Agents SDK, LangChain, LangGraph, N8N, MCP) для создания работающих агентов и мультиагентных систем.
- Научиться применять мультиагентные подходы для решения реальных задач ИИ, включая интеграцию с внешними сервисами, управление памятью и состоянием, а также оценку эффективности.

Задачи преподавателя:

- Обеспечить студентов структурированными лабораторными работами, охватывающими спектр методов от классических до современных интеллектуальных.
- Предоставить доступ к необходимым программным средствам и вычислительным ресурсам (Python, библиотеки, среды для визуализации).
- Организовать проверку и обратную связь по выполненным работам, акцентируя внимание на корректности реализации, анализе результатов и качестве выводов.

Ожидаемые результаты студентов:

- Научиться применять и разрабатывать интеллектуальных агентов
- Настраивать агентов и управлять их контекстом и задачами
- Интегрировать агентов с внешними сервисами
- Создавать обученных интеллектуальных агентов, способных решать частные задачи ИИ и координировать свою работу с другими агентами
- Реализовывать готовые VLA-модели (RT-2, GATO) для решения стандартных задач на стыке зрения, языка и действий. Применять техники prompt engineering для адаптации моделей к конкретным сценариям. Интегрирует предобученные эмбединги текста и изображений в pipeline принятия решений агента.
- Выполнять обучение интеллектуальных агентов на основе имеющихся данных

4.5 Методические указания по организации проектной деятельности студентов

Условия применения:

Курс рассчитан на студентов 4-го года обучения,
Имеется доступ к кейсам индустриальных партнеров.

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

Дать представление о реальных задачах, решаемых с помощью мультиагентных систем и возникающих проблемах.

Задачи преподавателя:

- сбор кейсов индустриальных партнеров;
- сбор кейсов преподавателей практиков и лабораторий в вузе;
- формирование ТЗ на зачетный проект на основе кейсов;
- разработка системы учёта результатов проекта в итоговой оценке.

Ожидаемые результаты студентов:

Начальное представление о реальных задачах, решаемых с помощью мультиагентных систем и возникающих проблемах.

Рассмотрим примеры кейсов.

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 приближенности 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.

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 правке 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. 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.

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

5.1 Основная литература:

1. Nathan Steele. Agentic AI with MCP: Build Structured Multi-Agent Systems with Model-Text Protocol Servers, LangChain, and Large Language Models (Building Intelligent Agents: The Complete Framework Series). 2025. ISBN 979-8287399009.
2. Rowan T. Mercer. MCP Server Development: A Hands-On Guide to Building Context-Aware AI Systems with OpenAI, Claude & LangChain. ISBN 979-8296682789. 2025.

5.2 Дополнительная литература:

- 1 Документация OpenAI SDK.

- 2 Документация LangChain.
- 3 Документация LangGraph.
- 4 Документация LangFlow.

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.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 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.ucheba.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 А*

Palayamkot, M. R., & Karim, K. (2025). MASON - A Multi-Agent LLM Framework for No-Code Development. Proceedings of the AAAI Symposium Series, 6(1), 313-322. <https://doi.org/10.1609/aaais.v6i1.36068>

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

По курсу предусмотрено проведение лекционных занятий, на которых дается систематизированный материал по агентным и мультиагентным системам. 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 разработки агентных и мультиагентных систем.

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

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. Перечень информационных технологий, используемых при осуществлении образовательного процесса 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, Python, Java.

8. Материально-техническое обеспечение по дисциплине (модулю)

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

№	Продукт	Параметры продукта	Кол-во	Кол-во конф игура ций	Ед. изм.
1	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	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.