

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

УТВЕРЖДАЮ:

Проректор по учебной работе,
качеству образования – первый проректор

_____ Хагуров Т.А.

подпись

«02» июня 2026



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.В.15 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ИИ

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) Искусственный интеллект и аналитика данных

Форма обучения очная

Квалификация бакалавр

Краснодар 2026

Рабочая программа дисциплины «Инструментальные средства разработки ИИ» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем.

Программу составили:

А.В. Коваленко, заведующий КАДИИ,
д-р.тех.н., профессор



подпись

Д.Н. Фокин
Генеральный директор ООО «Кар Икс Технолджис»



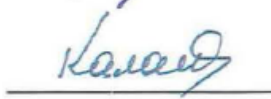
подпись

Е.Н. Калайдин, доктор физ.-мат.наук,
профессор



подпись

Г.В. Калайдина, канд. физ.-мат.наук,
доцент



подпись

Рабочая программа дисциплины утверждена на заседании центра
искусственного интеллекта
протокол № 4 «14» мая 2026 г.
Руководитель центра ИИ Коваленко А.В.



подпись

Утверждена на заседании учебно-методической комиссии факультета
компьютерных технологий и прикладной математики
протокол № 3 «15» мая 2026 г.
Председатель УМК факультета Добровольская Н.Ю.



подпись

Экспертиза:

Экспертная группа индустриального партнера ООО «CarX Technologies»:
Фокин Д.Н., Генеральный директор, seo@carx-tech.com; Щеголев А.В.,
Ведущий инженер-разработчик, a.shchegolev@carx-tech.com; Болоцкий Д.Д.,
DevOps-инженер, d.bolotskiy@carx-tech.com.

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
ФГБОУ ВО «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

Целью освоения дисциплины является формирование у студентов комплекса теоретических знаний и практических компетенций в области применения современных инструментальных средств разработки интеллектуальных агентов на основе генеративных моделей (LLM), включая умение проектировать, реализовывать и интегрировать агентов с использованием API-вызовов, RAG-архитектур, многоагентных паттернов и внешних сервисов.

В результате обучения студент должен быть способен самостоятельно создавать ИИ-агентов для решения прикладных задач в профессиональной деятельности, используя облачные платформы (Yandex Cloud AI) и подходы промпт-инжиниринга, вызова инструментов (Function Calling), управления памятью и семантического поиска.

Изменения связанные с включение в рабочую программу дисциплины материалов индустриального партнера помечены аббревиатурой МИП

Аббревиатура МИП – «Материалы индустриального партнера»

1.2 Задачи дисциплины

- Освоить базовые интерфейсы и модели облачной платформы – изучить возможности Yandex AI Studio, Responses API, модели YandexGPT и YandexART, научиться настраивать контекст диалога через `previous_response_id`.
- Сформировать навыки промпт-инжиниринга – освоить структуру эффективных промптов (системные инструкции, ограничения, форматы вывода), стратегии zero-shot и few-shot, приёмы управления выводом (ролевые подсказки, JSON-схемы), а также методы снижения галлюцинаций.
- Научиться проектировать и реализовывать ИИ-агентов – создать класс агента-обёртки над LLM, организовать агентный цикл с чередованием рассуждения и действий (ReAct), реализовать многошаговые сценарии с остановкой по условию.
- Овладеть механизмами интеграции внешних инструментов – использовать Function Calling для подключения пользовательских функций (анализ данных, визуализация, работа с файлами), обрабатывать ошибки вызовов и управлять правами доступа.
- Изучить архитектуру RAG (Retrieval-Augmented Generation) – освоить создание векторных хранилищ, семантический поиск по документам и файлам, интеграцию инструментов `file_search` и `web_search` в агента, научиться оптимизировать чанкинг и качество поиска.
- Освоить протокол MCP (Model Context Protocol) – научиться подключать готовые MCP-серверы (погода, базы данных, научные репозитории) и создавать собственные MCP-серверы через MCP Hub Yandex Cloud без кода.
- Приобрести навыки построения многоагентных систем – реализовать паттерны координации (sequential chain, coordinator, agents-as-tools), использовать OpenAI Agents SDK, настраивать мониторинг через хуки и события, управлять состоянием и долговременной памятью агентов.
- Закрепить компетенции через практическую проектную деятельность – выполнить цикл лабораторных работ (11 тем, от простого LLM-театра до медицинского deep research агента) и (факультативно) курсовой проект по одной из 40 предложенных тем, демонстрирующий интеграцию изученных инструментов для решения реальной прикладной задачи.
- **МИП:** Сформировать практические навыки прототипирования MVP игрового проекта с использованием генеративного ИИ (аудио, видео, код, ИИ-помощники) и реализации ML-агента, обучаемого с подкреплением в динамической среде (гоночный сценарий «догони игрока»), интегрируя полученные знания с существующими инструментами (RAG, Function Calling, MCP).

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Инструментальные средства разработки ИИ» относится к «Части, формируемой участниками образовательных отношений» Блока 1 «Дисциплины (модули)» учебного плана.

Пререквизиты: (Инфраструктурные дисциплины) Основы программирования, Алгоритмы и структуры данных, Введение в специальность (Темы: 3. Принципы промпт-инжиниринга, 4. Шаблоны и стратегии промптов), Фундаментальные дискретные модели, Математический анализ, Алгебра и геометрия.

Постреквизиты:

Б1.В.13 Разработка ИИ агентов (LLM-4.1(С)), Б1.В.08 Методы обучения с подкреплением (LLM-4.1(П)), Б1.В.ДВ.01.01 Мультиагентные системы (LLM-4.1(П)), Б1.В.ДВ.01.02 Инженерия интеллектуальных агентов (LLM-4.1(П)), Б1.В.ДВ.01.01 Мультиагентные системы (LLM-4.2(П)), Б1.В.ДВ.01.02 Инженерия интеллектуальных агентов (LLM-4.2(П))

Б1.В.06 Современные методы фронтальных исследований ИИ (LLM-5.3(П), LLM-5.4(П)), Б1.В.ДВ.01.01 Мультиагентные системы (LLM-5.4(П)), Б1.В.ДВ.01.02 Инженерия интеллектуальных агентов (LLM-5.4(П))

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: Data Analyst (Аналитик данных)

Извлечение знаний из данных, построение аналитических моделей, использующих МО и ИИ.

Задачи:

- 1. Статистический анализ, визуализация данных, предварительная обработка.*
- 2. Создание прогнозных моделей*
- 3. Построение аналитических моделей для поддержки бизнес-решений.*

Роль 2: MLOps (Специалист по эксплуатации ИИ)

Автоматизация и операционное управление жизненным циклом МО-моделей

Задачи:

- 1. DevOps для ML.*
- 2. Автоматизация, мониторинг ML-систем.*
- 3. Операционное управление жизненным циклом ML-моделей.*

Роль 3: AI PM (Менеджер проектов ИИ)

Управление процессами создания ИИ-решений, включая координацию команды разработки

Задачи:

- 1. Управление ИИ-проектами от идеи до внедрения*
- 2. Анализ бизнес-требований и постановка задач*
- 3. Оценка эффективности и ROI ИИ-решений*

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:

Код, уровень и формулировка компетенции	Индикаторы	Уровни освоения индикаторов компетенции
LLM-4 Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей	LLM-4.1. Умеет применять и разрабатывать интеллектуальных агентов	Знает: – архитектуру агента (модель, инструкции, память) паттерны ReAct, планирование-и-выполнение

		– (МИП) методы генерации контента (аудио, видео, код) с помощью LLM для игровых задач.
		Умеет: – создавать простого агента с одним инструментом – настраивать системный промпт – (МИП) использовать ИИ-помощников для создания работающего прототипа в среде Unity.
		Владеет: – разворачивать агента в Yandex Cloud – отлаживать агентный цикл – (МИП) приёмами интеграции сгенерированного ИИ контента в игровой движок.
	LLM-4.2. Интегрирует агентов с внешними сервисами	Знает: – протоколы API (REST, OpenAI-совместимый) – методы аутентификации (API-ключи, IAM)
		Умеет: – писать код-обёртку для внешнего сервиса – передавать параметры через Function Calling
		Владеет: – подключать агента к БД, поисковому API, файловому хранилищу – обрабатывать ошибки сервисов
	LLM-4.3. Разрабатывает агентные паттерны	Знает: – виды паттернов (sequential, coordinator, agents-as-tools) – преимущества многоагентных систем – (МИП) принципы RL (агент, среда, действия, награда) для задачи преследования.
		Умеет: – реализовывать цепочку из двух агентов – использовать агент как инструмент другого агента – (МИП) настраивать среду симуляции автомобиля и обучать агента политике «догони игрока».
		Владеет: – проектировать многоагентную систему под задачу – применять OpenAI Agents SDK – (МИП) методами отладки и визуализации обучения RL-агента в реальном времени.
	LLM-4.4. Управляет состоянием и памятью агентов	Знает: – типы памяти (кратко- и долгосрочная) – механизм previous_response_id и буфер сообщений
		Умеет: – сохранять историю диалога – использовать системный промпт для задания личности
		Владеет: настраивать постоянную память через внешнее хранилище

		– управлять контекстом в длительных сессиях
LLM-5 Организует взаимодействие с генеративными моделями через проектирование, анализ и применение промптов	LLM-5.3 Настраивает API для работы с LLM	Знает: – параметры вызова (температура, top-P, max_tokens)
		Умеет: – создавать клиент к Yandex Cloud / OpenAI API – обрабатывать потоковый ответ
		Владеет: – оптимизировать параметры для задач (снижение галлюцинаций) – логировать и анализировать запросы
	LLM-5.4 Разрабатывает дизайн и структуру промптов	Знает: – элементы эффективного промпта (роль, инструкция, ограничения, примеры) – приемы structured output (JSON-схема)
		Умеет: – писать системные и пользовательские промпты для задач суммаризации, извлечения фактов – использовать zero-shot и few-shot стратегии
		Владеет: – создавать шаблоны промптов для повторного использования – валидировать вывод через Pydantic

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

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

Вид учебной работы	Всего часов	Семестры (часы)
		3
Контактная работа, в том числе:	57,2	57,2
Аудиторные занятия (всего):	50	50
Занятия лекционного типа	16	16
Лабораторные занятия	34	34
Занятия семинарского типа (семинары, практические занятия)		
Иная контактная работа:	7,2	7,2
Контроль самостоятельной работы (КСР)	2	2
Промежуточная аттестация (ИКР)	5,2	5,2
Самостоятельная работа, в том числе:	14,8	14,8
Подготовка курсовой работы	5	5
Проработка учебного (теоретического) материала	5	5
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)	4,8	4,8
Реферат		
Подготовка к текущему контролю		
Контроль:		
Подготовка к экзамену		
Общая трудоемкость	час.	72
	в том числе контактная работа	57,2
	зач. ед	2

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы (темы) дисциплины, изучаемые в 3 семестре

№	Наименование разделов (тем)	Количество часов			
		Всего	Аудиторная работа		Внеаудиторная работа
			Л	ЛР	СРС
1	2	3	4	6	7
1.	Введение в YANDEX AI STUDIO	3	0	2	1
2.	Введение в ИИ-агенты	6	1	4	1
3.	Создание агента.	6	1	4	1
4.	Вызов внешних инструментов	7	2	4	1
5.	Подключение внешнего источника знаний (RAG)	8	2	4	2
6.	Вызов инструментов в объектно-ориентированном стиле	3	1	-	2
7.	Инструменты внешнего поиска	7	1	4	2
8.	МСП-серверы и их использование	8	2	4	2
9.	Многоагентные системы	8,8	2	4	2,8
10.	(МИП.1) Прототипирование MVP игрового проекта с помощью ИИ	4	2	2	
11.	(МИП.2) ML-агент с обучением с подкреплением в среде «догони игрока»	4	2	2	
ИТОГО по разделам дисциплины		64,8	16	34	14,8
Контроль самостоятельной работы (КСР)		2			
Промежуточная аттестация (ИКР)		5,2			
Подготовка к текущему контролю					
Общая трудоемкость по дисциплине		72			

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

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Введение в YANDEX AI STUDIO	Что такое Yandex AI Studio. Способы работы с AI Studio. Модели YandexGPT и YandexART. Responses API и диалоговый контекст.	Тест
СОЗДАНИЕ ИИ-АГЕНТОВ В YANDEX CLOUD			
2.	Введение в ИИ-агенты	Агенты и окружение. Характеристики агентов. Анатомия простейшего агента. Структурный ответ и вызов инструментов. Фундаментальные модели в Yandex Cloud. Фреймворки для создания агентов Многоагентные системы и многоагентные сценарии.	Тест
3.	Создание агента.	Начало работы с языковыми моделями в облаке Использование LLM	Тест

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		Responses API и история переписки	
4.	Вызов внешних инструментов	Что такое Function Calling и зачем он нужен. Как работает Function Calling. Реализация Function Calling для фитнес-ассистента. Определение функции для записи упражнений. Описание функции для языковой модели в Responses API. Вызов модели с инструментом. Обработка ответа модели. Function Calling (иллюстрация на лекции): Агент-исследователь научных статей. Знакомство с arXiv API. Настройка OpenAI клиент. Определение функций для Function Calling. Реализация функций. Диспетчер вызовов функций. Полный цикл работы с агентом. Использование Pydantic для описания функций. Класс Agent. Создание агента-исследователя.	Тест
5.	Подключение внешнего источника знаний (RAG)	Дообучение моделей и In-Context Learning RAG-архитектура. Как осуществляется поиск. Семантический поиск и векторные базы знаний. Разбиение текста на фрагменты. Добавление инструмента поиска в ассистента. Оптимизация RAG. Расширение RAG-конвейера.	Тест
6.	Вызов инструментов в объектно-ориентированном стиле	Использование Pydantic-объектов	Тест
7.	Инструменты внешнего поиска	Зачем нужен внешний поиск. Инструмент веб-поиска. Учёт истории переписки. Кастомизация инструмента поиска. Особенности использования инструмента поиска.	Тест
8.	MCP-серверы и их использование	Виды MCP-серверов. Механика вызова MCP. Как реализовать свой MCP-сервер. Использование MCP-сервера. MCP Hub в Yandex Cloud. MCP-сервер на Cloud Functions. Использование MCP-серверов в фитнес-ассистенте.	Тест
9.	Многоагентные системы	От инструментов к агентскому циклу. OpenAI Agents SDK. Мониторинг: RunHooks. Мониторинг: Streaming Events Многоагентные системы: зачем нужно несколько агентов. Паттерны координации многоагентных систем. Паттерн B: Sequential Chain (Последовательная цепочка). Паттерн "Координатор".	Тест

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		Паттерн A: Agents-as-Tools (Агенты как инструменты).	
10.	МИП: Прототипирование MVP Игрового проекта с помощью инструментов ИИ	Использование ИИ для развития концепции игрового проекта до состояния «минимально необходимой ценности продукта» (MVP). Применение генеративного ИИ для создания аудио-, видео-контента, ИИ-помощников (GitHub Copilot, Codeium) для генерации кода прототипа в Unity/Unreal. Пример: автоматическая сборка сцены, скриптов движения, UI через промпты. Оценка MVP: функциональность, качество контента, время разработки.	Тест + кейс
11.	МИП: Реализация ML-агента, обучаемого с подкреплением в среде «догони игрока»	Постановка задачи преследования (car following / pursuit). Обучение одного или больше ML-агентов (гоночный автомобиль) для прохождения трассы в соперничестве с другим агентом/игроком. Пространство состояний (скорость, угол, расстояние до цели), действия (газ, тормоз, руль), функция награды (приближение, удержание в зоне, финиш). Использование библиотек ML-Agents (Unity) или стабильных RL-фреймворков. Демонстрация обучения: от хаотичного движения до контролируемого преследования.	Тест + кейс

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

2.3.2 Занятия семинарского типа

Не предусмотрены

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Введение в YANDEX AI STUDIO	Диалог двух LLM-агентов с разными системными.	ЛР
2.	Введение в ИИ-агенты	Создание фитнес-ассистента.	ЛР
3.	Создание агента.	Агентный цикл генерации изображений. Суммаризация научной статьи.	ЛР
4.	Вызов внешних инструментов	Function Calling: Агент-аналитик с построением графиков. Function Calling: Агент-художник с Turtle Graphics	ЛР
5.	Подключение внешнего источника знаний (RAG)	RAG: Поиск в интернете (Web Search) RAG: Поиск по файлам (File Search)	ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
7.	Инструменты внешнего поиска	RAG Lab: Поиск по базе научных статей OpenAlex	ЛР
8.	МСП-серверы и их использование	MCP Lab: Туристический советник с погодным MCP-сервером)	ЛР
9.	Многоагентные системы	Medical Deep Research Agent	ЛР
10.	МИП: Прототипирование MVP Игрового проекта с помощью инструментов ИИ	1. Выбор концепции (2D-платформер, гонки на выживание, сбор предметов). 2. Использование LLM (YandexGPT, ChatGPT) для генерации: GDD (дизайн-документа), скриптов движения, UI, звуковых эффектов (через API генерации аудио). 3. Генерация текстур/спрайтов через YandexART или Stable Diffusion. 4. Сборка работающего прототипа в Unity (или Godot, Pygame) с интеграцией сгенерированных ассетов и кода. 5. Презентация MVP: управление, цель, минимальная игровая механика.	ЛР
11.	МИП: Реализация ML-агента, обучаемого с подкреплением в среде «догони игрока»	1. Создание упрощённой 2D/3D среды (Unity ML-Agents или пользовательская среда на Python + PyGame). 2. Определение агента-преследователя (следует за целевым агентом/игроком). 3. Реализация цикла RL: действия (ускорение, руль), наблюдения (относительная позиция, скорость), награда за удержание дистанции или столкновение. 4. Обучение с использованием PPO (например, через stable-baselines3 или Unity ML-Agents). 5. Визуализация обучения: график награды, поведение в реальном времени. 6. Финальное тестирование обученного агента против простого ИИ-игрока.	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)

1. Разработка агента-репетитора для подготовки к экзамену по математическому анализу (RAG + Function Calling).
2. ИИ-ассистент для составления индивидуальных планов тренировок (фитнестрекер с историей и расчётом калорий).
3. Многоагентная система для анализа новостей ИИ-индустрии (Web Search + суммаризация + PDF-отчёт).
4. Агент-художник с итеративным улучшением изображений на основе обратной связи VLM.

5. Система для автоматического реферирования научных статей из arXiv (RAG + структурированный вывод).
6. Агент-аналитик для визуализации данных из текстовых отчётов (matplotlib + Function Calling).
7. Туристический советник с интеграцией погодного MCP-сервера и поиска достопримечательностей.
8. Медицинский deep research агент с поиском по medRxiv и генерацией отчёта PDF.
9. Telegram-бот для семантического поиска по корпоративной базе знаний (File Search + RAG).
10. Агент для автоматического тестирования кода на основе описания функций на естественном языке.
11. Система «LLM-театр» с тремя и более агентами, ведущими дискуссию по заданной теме.
12. Агент-переводчик технической документации с сохранением стиля и терминологии.
13. Инструмент для извлечения сущностей (NER) из научных текстов с выводом в JSON Schema.
14. Агент для планирования расписания и управления задачами (календарь + напоминания).
15. Система вопросно-ответного поиска по базе ГОСТ и СНиП (RAG с чанкингом).
16. Многоагентная система для модерации контента (проверка токсичности, фактчекинг).
17. Агент-консультант по подбору книг на основе анализа рецензий (Web Search + сентимент).
18. Автоматический генератор учебных тестов по лекциям (извлечение фактов + генерация вариантов).
19. Агент-исследователь динамики цен на товары с построением графиков.
20. Система для сравнения двух научных статей (суммаризация + таблица различий).
21. Агент-парсер и классификатор вакансий с интеграцией в ATS.
22. ИИ-ассистент для ведения переговоров на основе заданной стратегии.
23. Многоагентная система для коллективного принятия решений (метод Дельфи).
24. Агент для преобразования «плохо структурированных» таблиц Excel в чистые данные.
25. RAG-бот по истории архитектуры с загрузкой изображений зданий.
26. Агент-рерайтер маркетинговых текстов с A/B-тестированием промптов.
27. Система мониторинга конкурентов: ежедневный отчёт о новых публикациях и продуктах.
28. Агент-тьютор для обучения работе с инструментами (Function Calling + объяснение шагов).
29. Автоматический суммаризатор лекций с выделением ключевых формул и определений.
30. Агент для поиска и связывания фактов в нескольких документах (граф знаний + RAG).
31. Система генерации художественных описаний к фотографиям с оценкой качества.
32. Агент для оптимизации расписания учебной группы с учётом пожеланий.
33. MCP-сервер для доступа к внутренним базам данных университета.
34. Агент-драматург: написание коротких сценариев с заданными персонажами.
35. Инструмент для проверки согласованности требований в техническом задании.

36. Агент-ассистент для проведения лабораторных работ по программированию (подсказки, проверка кода).
37. Система для детекции и исправления гендерных и иных стереотипов в текстах.
38. Многоагентная система для координации нескольких LLM-писателей (главы книги).
39. Агент для предсказания временной сложности алгоритма по псевдокоду.
40. Комплексный инструмент «Агентный конструктор» – GUI для сборки цепочек из готовых агентов.
41. (МИП) Разработка MVP гоночной аркады с использованием генеративного ИИ (трассы, автомобили, звук) и ручной доработкой в Unity.
42. (МИП) Создание обучаемого RL-агента для задачи «догони и удерживайся за лидером» в кастомной физической среде на Python/Box2D.
43. (МИП) Сравнение эффективности промпт-инжиниринга vs дообучения малой LLM для генерации игровых диалогов и квестов в RPG-прототипе.
44. (МИП) Многоагентная система (Agents-as-Tools) для автоматического тестирования сбалансированности игрового уровня: агент-гонщик + агент-аналитик + агент-балансир.

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 УМК ФКТиПМ, протокол №1 от 29.08.2025
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные УМК ФКТиПМ, протокол №1 от 29.08.2025
3.	Выполнение МИП-лабораторных работ (прототипирование, RL-агент)	Методические указания к лабораторным работам МИП, размещённые в корпоративном репозитории CarX TECHNOLOGIES (доступ по запросу). Рекомендованные tutoriales: Unity ML-Agents, документация по YandexART для генерации текстур, примеры промптов для генерации игрового кода.
4.	Подготовка курсового проекта по темам МИП	Рекомендации по оформлению игровых прототипов и обучению RL-агентов: структура отчёта, требования к видео-демонстрации, критерии оценки MVP.

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
1	ЛР	Практические 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 «Введение в специальность».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме тестовых заданий, кейсов и **промежуточной аттестации** в форме вопросов и заданий к **зачету**.

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

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

– 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	Код и наименование индикатора (в соответствии с п. 1.5)	Результаты обучения (в соответствии с п. 1.5)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	LLM-4.1. Умеет применять и разрабатывать интеллектуальных агентов	Знает: архитектуру агента (модель, инструкции, память); паттерны ReAct, планирование-исполнение. Умеет: создавать простого агента с одним инструментом; настраивать системный промпт. Владеет: разворачивать агента в Yandex Cloud; отлаживать агентный цикл. МИП: Знает: методы генерации контента (аудио, видео, код) для игр. Умеет: использовать ИИ-помощников для прототипирования в Unity.	Тест (вопросы №3, 5) Лабораторные работы №1, 2 Кейс №1 (фитнес-ассистент) МИП: Тест (вопросы №31, 34), Лабораторная работа №12, Кейс №5	Вопросы к зачету №3, 4, 5, 9 Практическое задание №4 (диалог агентов) МИП: Вопросы к зачету №31, 34; Практическое задание №6

		Владеет: интеграцией сгенерированного контента в движок.		
2	LLM-4.2. Интегрирует агентов с внешними сервисами	Знает: протоколы API (REST, OpenAI-совместимый); методы аутентификации. Умеет: писать код-обёртку для внешнего сервиса; передавать параметры через Function Calling. Владеет: подключать агента к БД, поисковому API, файловому хранилищу; обрабатывать ошибки сервисов.	Тест (вопросы №6,7,8,9,10,11,12,20,22,23,26,27,28) Лабораторные работы №2 (Function Calling), №5, №6, №7, №8, №9, №10, №11 Кейсы №1, №2, №4	Вопросы к зачету №7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19 Практические задания №1 (калькулятор), №2 (RAG), №5 (обработка ошибок)
3	LLM-4.3. Разрабатывает агентные паттерны	Знает: виды паттернов (sequential, coordinator, agents-as-tools); преимущества многоагентных систем. Умеет: реализовывать цепочку из двух агентов; использовать агент как инструмент другого агента. Владеет: проектировать многоагентную систему под задачу; применять OpenAI Agents SDK. МИП Знает: принципы RL (агент, среда, награда) для задачи преследования. Умеет: настраивать среду симуляции и обучать агента политике «догони игрока». Владеет: отладкой и визуализацией RL-обучения.	Тест (вопросы №5,13,14,21,29) Лабораторные работы №3 (агентный цикл), №11 (Medical Deep Research Agent) Кейс №3 (модерация контента) МИП: Тест (вопросы №32, 33, 35), Лабораторная работа №13, Кейс №6	Вопросы к зачету №9, 20, 21, 22, 28 Практическое задание №4 (диалог агентов) МИП: Вопросы к зачету №32, 33, 35; Практическое задание №7
4	LLM-4.4. Управляет состоянием и памятью агентов	Знает: типы памяти (кратко- и долгосрочная); механизм previous_response_id и буфер сообщений. Умеет: сохранять историю диалога; использовать системный промпт для задания личности. Владеет: настраивать постоянную память через внешнее хранилище; управлять контекстом в длительных сессиях.	Тест (вопросы №4,24) Лабораторные работы №1 (previous_response_id), №2 (история переписки)	Вопросы к зачету №5, 6, 24, 29 Практическое задание №4 (управление контекстом в диалоге)
5	LLM-5.3. Настраивает API для работы с LLM	Знает: параметры вызова (температура, top-P, max_tokens). Умеет: создавать клиент к Yandex Cloud / OpenAI API; обрабатывать потоковый ответ. Владеет: оптимизировать параметры для задач (снижение галлюцинаций); логировать и анализировать запросы.	Тест (вопросы №1,2,19,25,30) Лабораторные работы №1 (настройка клиента), №2, №3, №4	Вопросы к зачету №1, 2, 6, 7, 25 Практические задания №1, №2, №3, №4, №5 (все требуют настройки API)
6	LLM-5.4. Разрабатывает дизайн и структуру промптов	Знает: элементы эффективного промпта (роль, инструкция, ограничения, примеры); приемы structured output (JSON-схема). Умеет: писать системные и пользовательские промпты для задач	Тест (вопросы №15,16,17,18) Лабораторные работы №4 (структурированная суммаризация), №9 (сравнение RAG/Tool Calling) Кейс №2 (патентный RAG – промпт для отказа от ответа)	Вопросы к зачету №15, 16, 17, 18, 23, 26 Практическое задание №3 (извлечение фактов с JSON-схемой)

		суммаризации, извлечения фактов; использовать zero-shot и few-shot стратегии. Владеет: создавать шаблоны промптов для повторного использования; валидировать вывод через Pydantic.		
--	--	---	--	--

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 кейсы по тематике лабораторных работ

Кейс 1. Фитнес-ассистент с нестандартными функциями

Контекст.

Вы работаете в стартапе, разрабатывающем персонализированного ИИ-ассистента для фитнес-тренеров. В MVP уже реализованы функции: запись упражнения, подсчёт калорий, история тренировок. Тренеры начали использовать ассистента и попросили добавить две новые возможности:

1. отслеживание водного баланса (выпить воды, напоминание каждые 2 часа);
2. расчёт оптимального веса для упражнения (жим лёжа, приседания) на основе последних 5 тренировок по формуле Бжицки (псевдоним).

При этом тренеры жалуются, что ассистент «забывает» контекст после нескольких вопросов, и иногда вызывает функцию записи с отрицательными значениями повторений.

Роль студента.

Разработчик ИИ-агента (middle-уровень), который должен доработать существующего агента.

Задание (кейс).

1. Определите, какие изменения нужно внести в системный промпт, чтобы ассистент проверял корректность входных данных (повторения > 0 , вода > 0).
2. Опишите две новые функции (Pydantic-модели) для работы с водой и расчёта веса. Укажите, какие параметры и описания нужны.
3. Предложите механизм сохранения кратковременной памяти (как ассистент не забывает, что пользователь тренируется последние 20 минут).
4. Напишите псевдокод обработчика вызова функции `calculate_optimal_weight(exercise_name: str)`, который загружает последние 5 записей из хранилища и вычисляет 90% от среднего значения.
5. Как агент должен обработать ситуацию, когда пользователь вводит отрицательное количество выпитой воды?

Ожидаемый результат:

- Доработанный класс агента (на уровне схемы) с новыми инструментами;
- Текстовое описание политики безопасности и обработки ошибок;
- Демонстрация диалога «тренер ↔ ассистент», где агент запрашивает уточнение при неверных данных.

Кейс 2. RAG-бот для отдела патентов

Контекст.

Крупная компания имеет базу из 5000 патентных документов (PDF) на русском языке. Отдел патентоведов тратит много времени на поиск аналогов и формулировку заключений. Руководство решило внедрить ИИ-агента с RAG, который должен:

- принимать вопрос на естественном языке (например: «Найдите патенты, где используется полимерный композит с температурой плавления выше 200°C»);
 - искать по векторной базе;
 - возвращать цитату из патента и ссылку на файл.
- Однако при тестировании выяснилось:
- Модель иногда выдумывает номера патентов (галлюцинации);
 - Длинные патенты (50+ страниц) разбиты на чанки слишком мелко, теряется смысл;
 - Вопросы, требующие численного сравнения («выше 200°C»), работают плохо.

Роль студента.

AI-инженер, отвечающий за построение RAG-пайплайна.

Задание (кейс).

1. Предложите стратегию чанкинга для патентных документов с сохранением метаданных (номер патента, дата, раздел).
2. Какие методы можно использовать для улучшения поиска по числовым условиям (фильтрация до или после семантического поиска)?
3. Как с помощью системного промпта и/или постобработки снизить галлюцинации (агент должен отвечать только по найденным чанкам)?
4. Опишите, как интегрировать веб-поиск (`web_search`) в качестве резервного источника, если семантический поиск нашёл менее 2 релевантных документов.
5. Предложите структуру ответа агента, которая обязательно включает ссылку на первоисточник и отказ от ответа при недостатке данных.

Ожидаемый результат:

- Конфигурация векторного хранилища с параметрами чанкинга;
- Промпт для агента с правилами отказа от ответа;
- Схема гибридного поиска (семантический + числовой фильтр).

Кейс 3. Многоагентная система для модерации контента

Контекст.

Социальная сеть запускает автоматическую модерацию пользовательских комментариев с помощью многоагентной системы. Требуется проверять: токсичность, спам, оскорбления и призывы к насилию. В системе три типа агентов:

- **Агент-фильтр** (быстро отвергает очевидные нарушения по ключевым словам);
 - **Агент-аналитик** (LLM с доступом к правилам сообщества, выдаёт вероятность нарушения 0..1);
 - **Агент-апелляция** (при спорном случае подключает человека через API).
- Проблемы:
- Агент-фильтр иногда помечает безобидные слова (ложные срабатывания).
 - Агент-аналитик работает медленно, его вызов стоит денег.
 - Заказчик хочет, чтобы система обучалась на исправлениях модератора.

Роль студента.

Архитектор многоагентной системы.

Задание (кейс).

1. Предложите паттерн координации трёх агентов (последовательная цепочка? параллельный вызов с голосованием?). Обоснуйте.
2. Опишите, как организовать память для хранения прецедентов (ошибочных срабатываний), чтобы постепенно улучшать агента-фильтр без дообучения LLM.
3. Разработайте протокол коммуникации между агентом-аналитиком и агентом-апелляцией (какие данные передавать человеку-модератору?).
4. Какие инструменты (`function calling`) нужно предоставить агенту-аналитику для обращения к правилам сообщества?

5. Предложите механизм остановки системы: если агент-фильтр дал уверенность > 0.95, можно не вызывать аналитика.

Ожидаемый результат:

- Диаграмма последовательности вызовов агентов;
- Промпт для агента-аналитика с JSON-схемой вывода;
- Код-схема цикла обработки одного комментария.

Кейс 4. Туристический советник с кастомным MCP-сервером

Контекст.

Туристическое агентство хочет запустить Telegram-бота — советника по путешествиям. Бот должен:

- Узнавать предпочтения пользователя (страна, бюджет, интересы).
- Искать популярные места через веб-поиск (Yandex Search).
- Проверять текущую погоду в выбранных городах через общедоступный API OpenWeatherMap.
- Формировать таблицу с рекомендациями: город, достопримечательности, погода, совет взять зонт/солнцезащитный крем.

Технически: веб-поиск и вызов MCP-сервера должны быть доступны через Responses API. Погодный API требует API-ключ, но его нельзя хранить в коде бота в открытом виде. OpenWeatherMap возвращает JSON с текущей температурой, влажностью, осадками.

Вы уже знаете, что в Yandex Cloud есть MCP Hub, позволяющий создать MCP-сервер для любого REST API без кода.

Роль студента.

Интегратор решений на Yandex Cloud.

Задание (кейс).

1. Опишите пошаговую инструкцию создания MCP-сервера для OpenWeatherMap через MCP Hub: какие поля нужно указать (URL, метод, параметры, заголовок авторизации, JSON-преобразование).
2. Как подключить созданный MCP-сервер к агенту в Responses API? (Назовите поля в конфигурации `tools`.)
3. В системном промpte агента нужно явно описать алгоритм: сначала веб-поиск → выбрать 3 города → для каждого вызвать MCP-инструмент `get_weather`. Как гарантировать, что агент не вызовет погоду до поиска?
4. Предложите обработку ошибки: если MCP-сервер вернул `{ "error": "city not found" }` (например, пользователь ввел «Санта-Барбара», а API требует латиницу), агент должен сам преобразовать название или запросить уточнение. Какое сообщение об ошибке нужно вернуть модели в `tool_result`?
5. Напишите ожидаемый финальный ответ агента для запроса: «Хочу в Европу на майские праздники, бюджет 1000 евро».

Ожидаемый результат:

- Конфигурация MCP-сервера (yaml/JSON);
- Промпт агента с чёткими правилами использования инструментов;
- Пример диалога с обработкой ошибки геокодинга.

Кейс 5 (МИП). Прототипирование MVP гоночной аркады с использованием генеративного ИИ

Контекст.

Студия CarX TECHNOLOGIES хочет быстро проверить новую концепцию уличной гоночной аркады с элементами тюнинга. Задача: за 2 недели создать минимально жизнеспособный продукт (MVP), который можно показать фокус-группе. В команде нет художников и звукорежиссёров — только разработчики, владеющие промпт-инжинирингом и

интеграцией ИИ. Разрешено использовать любые генеративные модели для создания ассетов и фрагментов кода.

Роль студента.

AI-инженер-прототипировщик, отвечающий за настройку пайплайна генерации контента и сборку прототипа в Unity.

Задание (кейс).

1. Разработайте промпты для генерации: (а) трех фоновых треков (дрифт, стрит-рейсинг, ночной город) через LLM (описание для аудио-модели), (б) набора текстур дорожного покрытия и авто через YandexART, (в) базовых скриптов управления автомобилем и ИИ противника на промпте «создай C# скрипт для следования по трассе с поворотами».
2. Опишите архитектуру сборки MVP: какие компоненты генерируются ИИ, какие пишутся вручную.
3. Предложите критерии оценки качества сгенерированного контента (например, соответствие стилю, частота ошибок в коде).
4. Напишите системный промпт для агента, который помогает отбирать лучшие варианты текстур из 10 сгенерированных.
5. Оцените временные затраты: сколько человеко-часов сэкономило использование ИИ по сравнению с традиционной разработкой.

Ожидаемый результат:

- Дорожная карта создания MVP с указанием used AI tools.
- Примеры промптов и сгенерированных результатов (скриншоты, фрагменты кода).
- Оценка эффективности (ROI) применения ИИ в прототипировании.

Кейс 6 (МИП). Обучение RL-агента для задачи «догони лидера» в физической симуляции

Контекст.

В игре CarX Street игроки жалуются на слишком предсказуемых противников. Нужен адаптивный AI-соперник, который учится преследовать лидера гонки, используя реальную физику автомобиля (инерция, сцепление с дорогой). Команда уже имеет симуляционную среду на Unity ML-Agents. Требуется обучить агента, который может догонять и удерживаться за целевым автомобилем на трассе с поворотами, не вылетая.

Роль студента.

ML-инженер, настраивающий среду, функцию награды и параметры обучения PPO.

Задание (кейс).

1. Опишите пространство наблюдений (observations) для агента: какие параметры (относительная скорость, угол, расстояние, текущая скорость, показания датчиков касания дороги) необходимо передавать.
2. Предложите функцию награды (reward function) на каждом шаге, которая поощряет: сокращение дистанции до лидера, нахождение в зоне "след в след" (tailgating), прохождение контрольных точек. И штрафует: вылет с трассы, столкновение, слишком большое отставание.
3. Какие действия (discrete/continuous) должны быть у агента: газ/тормоз/руль. Обоснуйте.
4. Опишите конвейер обучения: как собирать опыт, как часто обновлять политику, как избегать переобучения под конкретную трассу.
5. Предложите метод оценки качества обученного агента (например, среднее время удержания лидера, количество обгонов за 5 кругов).
6. Напишите псевдокод цикла сбора данных для обучения с подкреплением (agent step → environment step → reward → buffer).

Ожидаемый результат:

- Спецификация среды и RL-параметров.

- Графики обучения (expected reward per episode).
- Видеодемонстрация поведения обученного агента (описание того, что должно быть видно).
- Рекомендации по развёртыванию модели в рантайме игры.

Кейсы покрывают основные темы курса: Function Calling, RAG, многоагентные системы, MCP-серверы. Каждый кейс может быть решён в течение 2–4 академических часов (лабораторное занятие) или использован как зачетное задание.

Лабораторные работы

Лабораторная работа №1 (тема 1.)

Необходимо создать простой эксперимент: два LLM-агента с разными системными промптами ведут диалог друг с другом. Можно назвать это простейшим LLM-театром.

Цель: Создать класс Agent — обёртку над Responses API с поддержкой контекста через previous_response_id; Зафиксировать силу системных промптов — одна и та же модель с разными инструкциями ведёт себя совершенно по-разному; Организовать диалог двух агентов — каждый агент получает реплику другого как входное сообщение.

План лабораторной работы:

- 1: Настройка клиента
- 2: Класс Agent
- 3: Определяем двух агентов
- 4: Функция диалога
- 5: Запускаем дискуссию!
- 6: Другие сценарии

Индивидуальное задание: создайте свою пару агентов.

Лабораторная работа №2 (тема 2, 3, 4)

Создание фитнес-ассистента

Создание класса для ассистента

Задание системного промпта и создание ассистента

Общение с ассистентом

Дополнительные задания:

Хранение истории переписки;

Изменение системного промпта;

Реализация Function Calling для фитнес-ассистента

1. Создайте набор функций для расширения возможностей ассистента. Для этого определите простое хранилище данных и три основных функции: запись упражнения, получение истории тренировок и расчёт сожжённых калорий
2. Опишите эти функции для языковой модели, чтобы она могла их вызывать. В поле description опишите каждую функцию, а внутри структуры parameters — полностью все параметры
3. Обновите класс Assistant из предыдущего урока — добавьте поддержку инструментов и их обработку
4. Создайте экземпляр ассистента с системным промптом, который описывает его роль и возможности

Дополнительные задания:

Добавьте функцию для отслеживания потребления воды;

Реализуйте функцию для подсчёта шагов;

Функция для отслеживания веса.

Лабораторная работа №3 (к теме Responses API)

Название: Агентный цикл генерации изображений.

Необходимо создать автоматический генератор изображений, основанный на агентном цикле, который будет итерационно улучшать генерируемое изображение на основе обратной связи от визуальной LLM.

Цель: Создать агентный цикл.

Задачи;

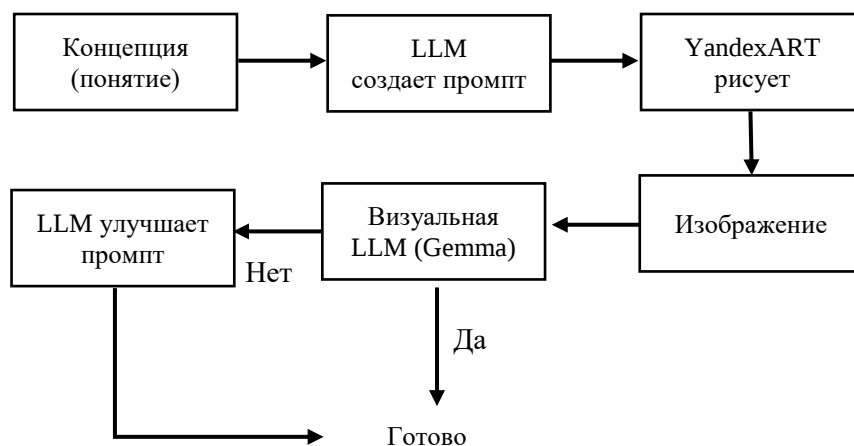
- Сгенерировать промпт с помощью YandexGPT на основе абстрактной концепции
- Создать изображение с помощью Yandex ART
- Оценит результат с помощью визуальной LLM (Gemma)
- Итеративно улучшить промпт на основе обратной связи

На вход подается простая концепция - *счастье* или *сложность*. На основе этой концепции:

- LLM генерирует промпт для рисования, продумывая, как лучше отобразить концепцию
- YandexART генерирует изображение
- Визуальная нейросеть VLM оценивает, насколько изображение соответствует концепции, и предлагает улучшения

После этого цикл повторяется.

Архитектура решения



План лабораторной работы:

- 1: Инициализация клиентов. Создание клиентов для работы с Yandex Cloud AI.
- 2: Вспомогательные функции. Создание функции для запроса к моделям генерации текста и изображений.
- 3: Добавляем структурный ответ. Для того, чтобы получать ответ от визуальной модели в структурированном виде, а также получить какой-то формальный критерий завершения итераций улучшения, будем использовать структурный ответ.
- 4: Агентный цикл. Задача - объединить все функции в агентный цикл, который будет итеративно улучшать изображение.
- 5: Эксперименты с разными концепциями. Проведите несколько экспериментов с разными концепциями и посмотрите на результаты. Представьте ответ на вопрос – помогает ли агентский цикл добиваться более качественных результатов в рисовании?

Лабораторная работа №4 (к теме Responses API)

Название: Суммаризация научной статьи.

Цель: Превратить неструктурированный научный текст в чистое, структурированное знание.

Необходимо взять классическую академическую статью "Attention Is All You Need" и используя языковые модели извлечь ее ключевые идеи превратив их в идеально организованное резюме.

Задачи:

- Очистка текста, извлечённого из PDF (исправление артефактов OCR)
- Извлечение структурированной информации из научной статьи.

План лабораторной работы:

1. Настройка окружения
2. Загрузка текста статьи. Извлечь текст статьи из PDF и сохранить в файле `article-attention.txt`. Такой текст часто содержит артефакты: разрывы слов, лишние переносы строк, остатки заголовков/колоннотитулов.
3. Очистка текста с помощью LLM. Используем LLM для исправления артефактов извлечения из PDF: Объединение разорванных слов; Удаление лишних переносов строк; Исправление опечаток и артефактов OCR.
4. Структурированная суммаризация. Продолжаем диалог с LLM (используем `previous_response_id`), предлагаем создать структурированное резюме статьи.
 - 4.1. Определение структуры данных.
 - 4.2. Генерация структурированного резюме с помощью Structured Outputs.
 - 4.3. Отображение результата.
5. Экспорт результатов. Сохраните результат суммаризации статьи в JSON-файл.

Дополнительные задания:

- Попробуйте сгенерировать вопросы для проверки понимания статьи.
- Сравните качество суммаризации разных моделей.

Лабораторная работа №5 (Function Calling: Агент-аналитик с построением графиков)

Цель: Создание инструмента для построения графиков, который агент может использовать для визуализации данных из текста.

Задачи:

Создать агента, который умеет:

- Анализировать текстовые данные (например, научную статью);
- Извлекать числовые данные из текста;
- Строить графики с помощью `matplotlib`.

Подсказка: LLM не может получить изображение обратно! Когда мы вызываем инструмент для рисования графика, мы можем сохранить изображение, но мы НЕ можем отправить его LLM как результат вызова функции. Поэтому мы используем паттерн глобального буфера:

- Создаём глобальный список `GRAPH_BUFFER`
- Каждый инструмент сохраняет `figure` в этот буфер
- После завершения работы LLM мы отображаем все графики из буфера

План лабораторной работы:

1. Настройка OpenAI клиента. Создадим клиент для работы с Yandex Cloud через OpenAI-совместимый API.
2. Глобальный буфер для графиков.
3. Определение инструментов для графиков.
4. Класс Agent для построения графиков.
5. Создание агента-аналитика.
6. Загрузка статьи "Attention Is All You Need".
7. Анализ ссылок по годам.

8. Сравнение BLEU Score моделей.
9. Анализ авторов и организаций.

Лабораторная работа №6 (Function Calling: Агент-художник с Turtle Graphics)

Цель: Научиться использовать Function Calling — механизм, позволяющий языковой модели вызывать внешние функции для выполнения задач.

Задачи:

Создать агента, который умеет рисовать с помощью библиотеки `jturtle` (Jupyter Turtle Graphics), со следующими функциями:

- Получать описание того, что нужно нарисовать (например, "дом").
- Вызывать функции движения черепашки: `forward`, `left`, `right`, `penup`, `pendown`.
- Создавать рисунок по своему "пониманию" задачи.

План лабораторной работы:

- 1: Знакомство с `jturtle`.
- 2: Настройка OpenAI клиента.
- 3: Определение функций для Function Calling.
- 4: Простой вызов с Function Calling.
- 5: Обработка вызовов функций.
- 6: Полный цикл рисования.
- 7: Использование Pydantic для описания функций.
- 8: Класс Agent. Создать универсальный класс Agent, который:
 - Автоматически генерирует описания инструментов из Pydantic-моделей;
 - Обрабатывает вызовы функций в цикле;
 - Управляет историей диалога.
- 9: Создание агента-художника. Создать агента с инструкцией для рисования.
- 10: Рисование с помощью агента. Функция-обёртка для удобного рисования.

Дополнительные задания:

- Добавьте новые команды черепашки (цвет, толщина линии).
- Создайте агента для работы с файлами или API.
- Создайте комбинацию несколько агентов.

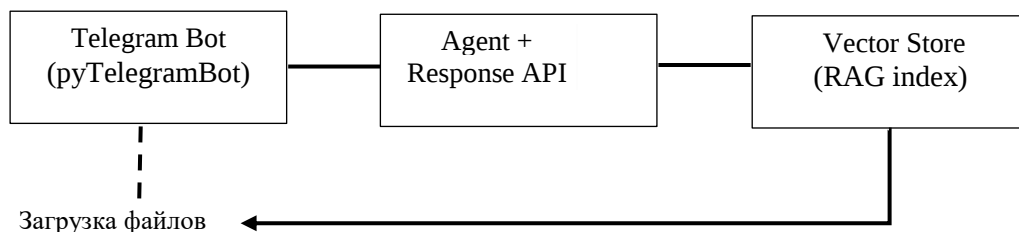
Лабораторная работа №7 (RAG: Поиск по файлам (File Search))

Создать персональную базу знаний из научных статей и сделать агента для работы с ними. Добавить интеграцию с Telegram-ботом для загрузки файлов и общения.

Задачи:

- Загрузить научные статьи в Yandex Cloud.
- Создать векторное хранилище для семантического поиска.
- Настроить агента с инструментом `file_search`.
- Интегрировать Telegram-бота для загрузки файлов и вопросов.

Архитектура



План лабораторной работы:

- 1: Импорт класса Agent и создание клиента.
- 2: Загрузка файлов в облако.
- 3: Создание векторного хранилища (Vector Store).
- 4: Проверка семантического поиска.
- 5: Создание агента с File Search.
- 6: Добавление новых файлов.
- 7: Telegram-бот для работы с базой знаний.
- 8: Очистка ресурсов.

Лабораторная работа №8 (RAG: Поиск в интернете (Web Search))

Цель: Создать агента, который использует инструмент Web Search для поиска актуальной информации в интернете.

Задачи:

- Научить наших агентов использовать инструмент веб-поиска web_search, чтобы они могли исследовать интернет (это может быть полезно во многих случаях для снижения галлюцинаций, получения актуальной информации и подведение итогов по поисковым выдачам).
- Создать сводку новостей об агентах искусственного интеллекта в 2025 году.

Архитектура



План лабораторной работы:

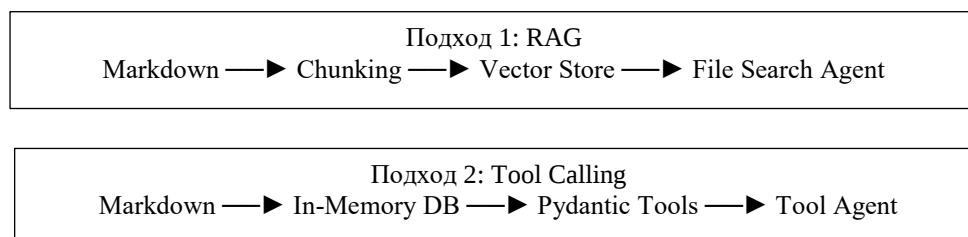
- 1: Импорт класса Agent
- 2: Настройка инструмента Web Search
- 3: Создание агента-исследователя
- 4: Исследование новостей об AI-агентах в 2025 году
- 5: Агент для мониторинга конкурентов

Лабораторная работа №9 (RAG Lab: Поиск по базе научных статей OpenAlex)

Цель: Сравнить два подхода к ответам на вопросы по базе данных научных статей из репозитория OpenAlex, представленной в виде Markdown-таблицы:

- 1) RAG (Retrieval-Augmented Generation) — семантический поиск по фрагментам таблицы с помощью векторного хранилища.
- 2) Tool Calling — структурированный поиск по in-memory базе данных через вызов функций.

Архитектура



План лабораторной работы:

1. Загрузка и исследование данных

2. Ручное чанкование и создание векторного хранилища
3. Проверка семантического поиска
4. Создание RAG-агента
5. Нетривиальные примеры с RAG
6. Tool Calling с Pydantic (in-memory база данных)
7. Сравнительная оценка: RAG vs Tool Calling

Рекомендации:

- Используйте RAG, когда важен поиск по смыслу и работа с большими объёмами неструктурированных данных
- Используйте Tool Calling, когда нужны точные вычисления, агрегация и структурированный доступ к данным
- В реальных приложениях часто комбинируют оба подхода для получения наилучших результатов.

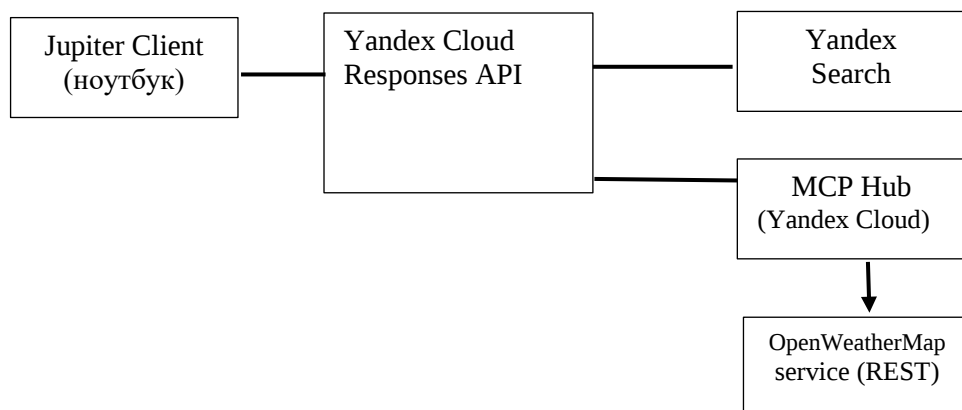
Лабораторная работа №10 (MCP Lab: Туристический советник с погодным MCP-сервером)

Цель: Создать AI-агента — туристического советника, который подбирает место для путешествия на основе предпочтений пользователя и актуальной погоды в городах-кандидатах.

Задачи: Агент

- Ищет в интернете подходящие направления (Web Search).
- Запрашивает текущую погоду через MCP-сервер (OpenWeatherMap).
- Формирует таблицу с рекомендациями.
- Responses API сам подключается к MCP-серверу и вызывает нужные функции!
- Для создания MCP-сервера из общедоступного OpenWeatherMap API используем MCP Hub Yandex Cloud - это получится сделать без единой строчки кода.

Архитектура



План лабораторной работы:

1. Настройка окружения и авторизация
2. Создание MCP-сервера для погоды
3. Настройка инструментов: Web Search + MCP
4. Создание агента-советника
5. Тестирование на разных запросах

Лабораторная работа №11 (Medical Deep Research Agent)

Цель:

Создать многоагентную систему для глубоких медицинских исследований, которая:

- Использует MCP-сервера для поиска научных статей (medRxiv, arXiv).
- Анализирует найденные материалы.
- Генерирует финальный отчет в формате PDF.

Задачи:

- Настроить подключение к Yandex Cloud и OpenAI Agents SDK.
- Подключить MCP-сервера для источников данных.
- Реализовать инструмент для компиляции/генерации PDF.
- Спроектировать архитектуру агентов (Writer, Researcher, Analyst и др.).
- Запустить исследование на медицинскую тему.

План лабораторной работы:

1. Инициализация и настройка.
2. Мониторинг (Hooks).
3. Подключение MCP-серверов.
4. Генерация отчета (PDF).
5. Инструмент заметок.
6. Архитектура агентов.
7. Запуск исследования.

Лабораторная работа №12 (МИП). Прототип своего игрового проекта в любой среде по выбору

Название: Прототип игрового проекта с генерацией контента через ИИ (MVP)

Цель: Сформировать практические навыки быстрого прототипирования игрового MVP с использованием генеративного ИИ (текст, изображения, аудио, код) и интеграции их в работающее приложение.

Задачи:

- Выбрать концепцию простой игры (платформер, гонки на выживание, сбор предметов).
- Сгенерировать с помощью LLM дизайн-документ и план реализации.
- Используя YandexART или аналоги, создать текстуры, спрайты, фоны.
- С помощью ИИ-помощника (GitHub Copilot, Codeium, или direct LLM API) сгенерировать код механик на выбранном движке (Unity C# / Pygame / Godot GDScript).
- Интегрировать сгенерированные ассеты и код в работающий прототип.
- Провести минимальное тестирование и подготовить демо-видео.

План лабораторной работы:

1. Подготовка: выбор идеи, написание системного промпта для генерации GDD.
2. Генерация ассетов: создание промптов для текстур и спрайтов, описание звукового ряда.
3. Генерация кода: запрос к LLM на создание скриптов управления, коллизий, UI. Адаптация под движок.
4. Сборка прототипа: импорт ассетов и кода в среду, настройка сцены.
5. Итеративное улучшение: при необходимости повторная генерация проблемных элементов.
6. Фиксация результатов: скриншоты, видео геймплея, репозиторий.
7. Защита: презентация MVP, ответы на вопросы о роли ИИ в разработке.

Лабораторная работа №13 (МИП). ML-агент на примере гоночной погони

Название: Реализация самообучающейся модели «преследования» с физической симуляцией автомобиля

Цель: Освоить базовые принципы обучения с подкреплением (RL) на примере задачи управления гоночным автомобилем в динамической среде.

Задачи:

- Создать упрощённую 2D или 3D среду с автомобилем, управляемым игроком (цель) и агентом-преследователем.
- Определить пространство наблюдений (позиции, скорости, расстояние до цели).
- Задать дискретные или непрерывные действия (газ/тормоз, угол поворота руля).
- Спроектировать функцию награды (reward function), поощряющую приближение к цели и удержание в зоне.
- Реализовать цикл обучения с использованием библиотеки stable-baselines3 (PPO) или Unity ML-Agents.
- Обучить агента до стабильного поведения «догони игрока».
- Визуализировать процесс обучения (графики награды, траектории).
- Сравнить эффективность агента до и после обучения.

План лабораторной работы:

1. Построение среды: создание физической симуляции (Pygame + Pymunk, или Unity с ML-Agents).
2. Интеграция RL-фреймворка: подключение stable-baselines3 или настройка ML-Agents.
3. Программирование награды: написание функции reward step (дистанция, выезд за границы, столкновение).
4. Запуск обучения: от 500 тыс. до 1 млн шагов, логирование.
5. Анализ результатов: построение графика средней награды, визуализация поведения чекпоинтов.
6. Тестирование обученного агента: запуск против простого ИИ или игрока (запись видео).
7. Оформление отчёта: описание гиперпараметров, кривых обучения, выводов.

Пример тестового задания для проведения текущей аттестации по теоретическому материалу лекций.

Разделы: лекции 1–9 (введение в Yandex AI Studio, агенты, function calling, RAG, MCP, многоагентные системы, промпт-инжиниринг)

Вопрос 1

Что такое Yandex AI Studio?

- a) Среда для ручного тестирования нейросетей
- b) Облачная платформа для работы с генеративными моделями YandexGPT и YandexART
- c) Библиотека для локального запуска LLM
- d) Система управления базами данных

Правильный ответ: b

Вопрос 2

Какой параметр в Responses API позволяет сохранять историю переписки между вызовами?

- a) session_id
- b) previous_response_id
- c) context_length
- d) message_history

Правильный ответ: b

Вопрос 3

Что из перечисленного является характеристикой ИИ-агента?

- a) Способность только генерировать текст
- b) Наличие автономности, реактивности, проактивности и социальной способности
- c) Обязательное наличие физического корпуса
- d) Использование только символьной логики

Правильный ответ: b

Вопрос 4

В чём заключается роль системного промпта (system prompt) в агенте?

- a) Он задаёт начальное сообщение пользователя
- b) Он определяет базовые инструкции, роль и ограничения поведения модели
- c) Он хранит историю вызовов инструментов
- d) Он кэширует векторные представления

Правильный ответ: b

Вопрос 5

Какой паттерн работы агента предполагает чередование шагов «рассуждение» и «действие» (вызов инструмента)?

- a) Планирование и выполнение (Plan-and-Execute)
- b) ReAct (Reasoning + Acting)
- c) Chain-of-Thought только без действий
- d) Одношаговый запрос-ответ

Правильный ответ: b

Вопрос 6

Для чего в агентах используется механизм Function Calling?

- a) Для вызова функций операционной системы
- b) Для того чтобы модель могла инициировать выполнение внешнего кода или API по структурированному описанию
- c) Для распараллеливания вычислений на GPU
- d) Для автоматической компиляции программ

Правильный ответ: b

Вопрос 7

Каким образом лучше всего описать функцию для Function Calling в коде на Python при работе с Yandex Cloud OpenAI-совместимым API?

- a) Написать docstring на русском языке в свободной форме
- b) Использовать класс Pydantic с полями и описаниями, а затем преобразовать в JSON Schema
- c) Передать модели просто имя функции без параметров
- d) Записать функцию в виде строки байт-кода

Правильный ответ: b

Вопрос 8

Что такое RAG (Retrieval-Augmented Generation)?

- a) Техника дообучения модели на новых данных
- b) Архитектура, дополняющая генерацию ответа поиском релевантных фрагментов из внешнего источника знаний
- c) Способ сжатия моделей
- d) Алгоритм кластеризации текстов

Правильный ответ: b

Вопрос 9

Для чего используется векторное хранилище (Vector Store) в RAG?

- a) Для хранения исходных PDF-файлов
- b) Для семантического поиска: хранения эмбеддингов фрагментов текста и быстрого поиска по косинусной близости
- c) Для кэширования ответов LLM
- d) Для логирования вызовов API

Правильный ответ: b

Вопрос 10

Какой инструмент в Yandex Cloud позволяет добавить агенту возможность поиска по загруженным файлам (семантический поиск)?

- a) web_search
- b) file_search
- c) code_interpreter
- d) image_generation

Правильный ответ: b

Вопрос 11

Что такое MCP (Model Context Protocol)?

- a) Протокол для обмена сообщениями между микросервисами
- b) Протокол для стандартизированного подключения внешних инструментов и источников данных к LLM-приложениям
- c) Язык разметки промптов
- d) Библиотека для визуализации внимания трансформеров

Правильный ответ: b

Вопрос 12

Какой сервис Yandex Cloud позволяет создавать MCP-серверы без написания кода, через конфигурацию?

- a) Yandex Data Proc
- b) MCP Hub
- c) API Gateway
- d) Object Storage

Правильный ответ: b

Вопрос 13

В многоагентной системе паттерн «Sequential Chain» означает:

- a) Агенты работают параллельно и конкурируют за ресурсы
- b) Агенты вызываются последовательно, каждый передаёт результат следующему
- c) Один главный агент координирует остальных через голосование
- d) Все агенты общаются через общую доску объявлений

Правильный ответ: b

Вопрос 14

Какой паттерн многоагентной системы предполагает, что один агент выступает как инструмент для другого агента (Agents-as-Tools)?

- a) Агент-координатор
- b) Агент-супервайзер
- c) Один агент может вызывать другого через Function Calling или специальный SDK
- d) Все агенты имеют одинаковый приоритет

Правильный ответ: с

Вопрос 15

Какая стратегия промптинга предполагает предоставление модели одного или нескольких примеров правильного ответа до выполнения основного задания?

- a) Zero-shot
- b) Few-shot
- c) Chain-of-Thought
- d) Role-playing

Правильный ответ: b

Вопрос 16

Для какой задачи лучше всего подходит zero-shot подход (без примеров)?

- a) Извлечение фактов из юридических документов по сложной схеме
- b) Преобразование стиля с технического на разговорный
- c) Общая суммаризация короткой новости, где формат очевиден
- d) Генерация SQL запросов по нестандартной схеме БД

Правильный ответ: с

Вопрос 17

Каким образом можно гарантировать, что LLM вернёт строго валидный JSON с заданной структурой?

- a) Попросить в свободной форме «верни JSON»
- b) Использовать параметр `response_format` с JSON-схемой или `structured_outputs` в API
- c) Поставить температуру 0
- d) Передать пример вывода в виде картинки

Правильный ответ: b

Вопрос 18

Что такое «ролевая подсказка» (role prompt) при проектировании промпта?

- a) Указание модели выполнить роль конкретного персонажа или эксперта
- b) Перечисление всех возможных ролей в компании
- c) Задание формата вывода в виде таблицы
- d) Ограничение длины ответа

Правильный ответ: а

Вопрос 19

Какой параметр API влияет на случайность (креативность) ответов модели?

- a) `max_tokens`
- b) `top_p` (или `temperature`)
- c) `frequency_penalty`
- d) `presence_penalty`

Правильный ответ: b

Вопрос 20

Какие данные передаются в сообщении типа `tool_result` при обработке Function Calling?

- a) Только статус выполнения (успех/ошибка)
- b) Результат выполнения функции (число, строка, JSON и т.д.) и опционально идентификатор вызова
- c) Исходный промпт пользователя

d) Системный промпт

Правильный ответ: b

Вопрос 21

Какая политика остановки агента является корректной для многошаговой задачи с неизвестным числом шагов?

a) Остановка после каждого шага без анализа

b) Остановка по достижении максимального числа итераций (например, 10) или когда модель возвращает специальный сигнал (`stop_reason = "end_turn"`)

c) Бесконечный цикл до принудительной остановки

d) Остановка после первого вызова инструмента

Правильный ответ: b

Вопрос 22

Как следует обрабатывать ошибку вызова инструмента (например, таймаут или неверные параметры) в агенте?

a) Завершить работу агента с ошибкой

b) Вернуть ошибку модели в виде структурированного сообщения `tool_result` и позволить модели скорректировать вызов

c) Игнорировать ошибку и продолжить

d) Перезапустить агента с начала

Правильный ответ: b

Вопрос 23

Какое ограничение безопасности необходимо накладывать на агента с доступом к файловому хранилищу?

a) Запретить чтение любых файлов

b) Разрешить доступ только к песочнице (определённой директории) и ограничить типы файлов

c) Никаких ограничений, агент сам разберётся

d) Передавать агенту пароли от системы

Правильный ответ: b

Вопрос 24

Что такое кратковременная память агента (short-term memory)?

a) База данных для долгосрочного хранения фактов

b) Контекст текущего диалога (сообщения пользователя, ответы модели, результаты вызовов инструментов)

c) Встроенные знания модели из обучения

d) Логирование всех действий агента на диск

Правильный ответ: b

Вопрос 25

Для чего используется `previous_response_id` в API Yandex Cloud?

a) Для пагинации ответов

b) Чтобы автоматически подхватить предыдущий контекст диалога без ручной передачи массива сообщений

c) Для идентификации версии модели

d) Для включения режима отладки

Правильный ответ: b

Вопрос 26

Какой из инструментов позволяет агенту получать актуальную информацию из интернета?

- a) file_search
- b) web_search
- c) vector_store
- d) code_interpreter

Правильный ответ: b

Вопрос 27

Что такое чанкинг (chunking) в контексте RAG?

- a) Сжатие изображений
- b) Разбиение большого текста на логические фрагменты (чанки) для последующего эмбединга и поиска
- c) Метод оптимизации промптов
- d) Процесс объединения нескольких моделей

Правильный ответ: b

Вопрос 28

В чём основное отличие RAG от дообучения (fine-tuning) модели?

- a) RAG не требует вычислений, а fine-tuning требует
- b) RAG добавляет поиск по внешним данным в момент инференса без изменения весов модели; fine-tuning меняет веса модели на новых данных
- c) RAG работает только с изображениями
- d) Fine-tuning быстрее RAG на порядок

Правильный ответ: b

Вопрос 29

Что такое агентный цикл (agent loop) в контексте генерации изображений с итеративным улучшением?

- a) Однократный вызов модели для рисования
- b) Повторяющаяся последовательность: генерация промпта → создание изображения → оценка качества (VLM) → улучшение промпта
- c) Цикл обучения модели на изображениях
- d) Конвейер загрузки файлов

Правильный ответ: b

Вопрос 30

Какая компетенция относится к умению настраивать API для работы с LLM и менять параметры вызова (температура, top_p)?

- a) LLM-4.1
- b) LLM-4.3
- c) LLM-5.3
- d) LLM-5.4

Правильный ответ: c

Вопрос 31 (МИП)

Что из перечисленного является примером использования генеративного ИИ для ускорения прототипирования игрового MVP?

- a) Оптимизация кода компилятором
- b) Генерация текстур, звуковых эффектов и скелетного кода по промпту
- c) Ручная отрисовка каждого спрайта
- d) Использование референсов из готовых игр

Правильный ответ: b

Вопрос 32 (МИП)

Какой метод наиболее эффективен для создания адаптивного ИИ-противника в гоночной игре с физикой?

- a) Детерминированный конечный автомат
- b) Обучение с подкреплением (RL) в симуляционной среде
- c) Жёстко прописанная логика следования по траектории
- d) Случайное блуждание

Правильный ответ: b

Вопрос 33 (МИП)

В задаче преследования лидера (car following) функция награды должна **штрафовать** за:

- a) Приближение к цели на дистанцию менее 2 метров
- b) Выезд за пределы трассы или столкновение с препятствием
- c) Удержание оптимальной дистанции
- d) Прохождение контрольной точки

Правильный ответ: b

Вопрос 34 (МИП)

При использовании ИИ для генерации кода в Unity, какой из следующих промптов даст наиболее полный и готовый к использованию скрипт?

- a) «Напиши скрипт для машины»
- b) «Создай C# скрипт для управления автомобилем с осями "Horizontal" и "Vertical", добавь силу двигателя, поворот колёс и торможение пробелом»
- c) «Дай код движения»
- d) «Сделай игру»

Правильный ответ: b

Вопрос 35 (МИП)

Какой фреймворк/библиотека чаще всего используется для обучения RL-агентов в игровых средах на Unity?

- a) TensorFlow
- b) PyTorch
- c) Unity ML-Agents
- d) OpenAI Gym

Правильный ответ: c

Таблица соответствия вопросов теста индикаторам компетенций

№ вопроса	Индикатор компетенции	Краткое обоснование соответствия
1	LLM-5.3	Настройка API – знание облачной платформы Yandex AI Studio
2	LLM-5.3	Настройка API – работа с параметром <code>previous_response_id</code>
3	LLM-4.1	Применение и разработка агентов – характеристики агента
4	LLM-4.4	Управление состоянием и памятью – системный промпт как часть состояния
5	LLM-4.3	Разработка агентных паттернов – паттерн ReAct
6	LLM-4.2	Интеграция с внешними сервисами – суть Function Calling

7	LLM-4.2	Интеграция с внешними сервисами – описание функции через Pydantic
8	LLM-4.2	Интеграция с внешними сервисами – архитектура RAG как способ подключения источника знаний
9	LLM-4.2	Интеграция с внешними сервисами – назначение векторного хранилища
10	LLM-4.2	Интеграция с внешними сервисами – инструмент file_search
11	LLM-4.2	Интеграция с внешними сервисами – знание протокола MCP
12	LLM-4.2	Интеграция с внешними сервисами – MCP Hub
13	LLM-4.3	Разработка агентных паттернов – паттерн Sequential Chain
14	LLM-4.3	Разработка агентных паттернов – паттерн Agents-as-Tools
15	LLM-5.4	Разработка дизайна и структуры промптов – стратегия few-shot
16	LLM-5.4	Разработка дизайна промптов – выбор zero-shot там, где он уместен
17	LLM-5.4	Разработка структуры промптов – structured output / JSON-схема
18	LLM-5.4	Разработка дизайна промптов – ролевая подсказка
19	LLM-5.3	Настройка API – параметры temperature / top_p
20	LLM-4.2	Интеграция с сервисами – обработка результата инструмента (Function Calling)
21	LLM-4.3	Разработка агентных паттернов – политика остановки агента
22	LLM-4.2	Интеграция с сервисами – обработка ошибок вызова инструментов
23	LLM-4.2	Интеграция с сервисами – ограничения безопасности при работе с файлами
24	LLM-4.4	Управление состоянием и памятью – кратковременная память
25	LLM-5.3	Настройка API – использование previous_response_id для контекста
26	LLM-4.2	Интеграция с внешними сервисами – инструмент веб-поиска
27	LLM-4.2 (через RAG)	RAG – чанкинг текста для семантического поиска
28	LLM-4.2	Интеграция с сервисами – понимание отличия RAG от дообучения
29	LLM-4.3 и LLM-4.4	Агентный цикл улучшения изображений (паттерн + управление памятью) – комбинированный индикатор, но в таблице отнесём к LLM-4.3 (паттерн)
30	LLM-5.3	Прямая проверка знания компетенции LLM-5.3
31	LLM-4.1. МИП (Разрабатывает прототип игрового MVP с помощью ИИ)	Проверка знания способов применения генеративного ИИ для создания игрового контента (текстуры, звук, код).
32	LLM-4.3. МИП (Реализует ML-агента с RL в физической среде)	Понимание области применения RL для адаптивного ИИ в динамических гоночных сценариях.
33	LLM-4.3. МИП (Реализует ML-агента)	Проверка умения правильно спроектировать функцию награды: что штрафовать (выход за границы) и что поощрять.

34	LLM-4.1. МИП (Разрабатывает прототип)	Оценка навыков промпт-инжиниринга для генерации рабочего кода на C# в Unity.
35	LLM-4.3. МИП (Реализует RL-агента)	Знание основного инструмента (Unity ML-Agents) для внедрения RL в игровые проекты.

- LLM-4.1 – вопрос 3, 31, 34 (3 вопроса)
- LLM-4.2 – вопросы 6,7,8,9,10,11,12,20,22,23,26,27,28 (13 вопросов)
- LLM-4.3 – вопросы 5,13,14,21,29,31,33, 35 (8 вопросов)
- LLM-4.4 – вопросы 4,24 (2 вопроса)
- LLM-5.3 – вопросы 1,2,19,25,30 (5 вопросов)
- LLM-5.4 – вопросы 15,16,17,18 (4 вопроса)

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет)

Вопросы для подготовки к зачету

1. Что такое Yandex AI Studio? Какие модели входят в её состав?
2. В чём разница между Responses API и Chat Completions API?
3. Что такое агент в контексте LLM? Основные характеристики агента.
4. Опишите анатомию простейшего агента (поток данных).
5. Что такое System Prompt и как он влияет на поведение модели?
6. Как организовать историю переписки с помощью `previous_response_id`?
7. Что такое Function Calling и для чего он нужен?
8. Как описать функцию для LLM с помощью Pydantic?
9. Какой паттерн работы агента следует выбрать для многошаговой задачи с неопределённым маршрутом? Почему?
10. Назовите три способа обработки ошибок при вызове инструментов агентом.
11. Какие ограничения безопасности нужно накладывать на агента с доступом к файлам?
12. Что такое RAG (Retrieval-Augmented Generation)? Из каких этапов состоит RAG-конвейер?
13. Для чего используется векторное хранилище? Как оно устроено?
14. Какие стратегии разбиения текста на фрагменты (chunking) вы знаете?
15. Чем семантический поиск отличается от полнотекстового?
16. Как подключить инструмент веб-поиска к агенту в Yandex Cloud?
17. Что такое MCP (Model Context Protocol)? Какие типы MCP-серверов существуют?
18. Опишите механизм вызова MCP-сервера из Responses API.
19. Что такое MCP Hub в Yandex Cloud? Как создать MCP-сервер без кода?
20. Назовите три паттерна координации многоагентных систем.
21. В чём разница между паттернами «Sequential Chain» и «Agents-as-Tools»?
22. Что такое агентный цикл улучшения изображений? Как в нём участвует VLM?
23. Как обеспечить структурированный вывод от LLM (JSON, таблица)?
24. Какие приёмы промпт-инжиниринга помогают снизить галлюцинации?
25. Когда лучше использовать zero-shot, а когда few-shot? Приведите примеры.
26. Что такое «ролевая подсказка» и как она влияет на результат?
27. Как сравнить качество суммаризации двух разных LLM?
28. Какие способы мониторинга многоагентных систем вы знаете (hooks, events)?

29. Как реализовать долговременную память агента с помощью внешней БД?
30. Перечислите этапы создания туристического агента с погодным MCP-сервером.
31. Опишите пайплайн создания MVP гоночной аркады с использованием генеративного ИИ: какие этапы можно автоматизировать, какие требуют ручной доработки?
32. (МИП) Какие особенности физической симуляции (инерция, сцепление, трение) необходимо учитывать при проектировании функции награды для RL-агента в задаче «догони игрока»?
33. (МИП) Сравните подходы к созданию игрового ИИ: детерминированный конечный автомат, поведенческие деревья и обучение с подкреплением. В каких сценариях RL даёт преимущество?
34. (МИП) Как с помощью промпт-инжиниринга можно улучшить качество генерируемого кода для Unity (C#) при создании прототипа? Приведите пример эффективного промпта.
35. (МИП) Какие метрики качества и времени разработки следует использовать для оценки эффективности применения генеративного ИИ при прототипировании игрового проекта?

Практические задания к зачету

Задания выполняются в среде Jupyter Notebook или Google Colab с доступом к Yandex Cloud AI.

Задание 1 (агент с инструментом)

Разработайте агента-калькулятора, который умеет вызывать функцию `calculate(expression: str) -> float`. Агент должен получить от пользователя выражение типа "сколько будет $(15 + 7) * 3$ " и вернуть вычисленный результат.

Демонстрирует: Function Calling, безопасную обработку вычислений.

Задание 2 (RAG на одном документе)

Создайте векторное хранилище из одного текстового файла (например, лекция в формате .txt). Реализуйте агента, который отвечает на вопросы по содержанию файла, используя инструмент `file_search`. Покажите два вопроса: прямого цитирования и требующего обобщения.

Демонстрирует: RAG, семантический поиск.

Задание 3 (структурированный вывод)

Напишите промпт и код для извлечения из новости (короткий текст) следующих полей в формате JSON:

`{"тема": "string", "дата_события": "date", "участники": ["string"], "ключевой_показатель": "число"}`.

Примените выход с JSON-схемой (structured output).

Демонстрирует: промптинг, форматирование вывода.

Задание 4 (многоагентный простой диалог)

Создайте двух агентов:

- **Агент А** – «Оптимист» (системный промпт: всегда находить плюсы в любой ситуации).
- **Агент Б** – «Реалист» (системный промпт: анализировать риски и трудности).
- Организуйте диалог из 4 реплик (А → Б → А → Б) на тему «Внедрение ИИ-агентов в университете». Выведите стенограмму.

Демонстрирует: класс Agent, управление контекстом, системные промпты.

Задание 5 (обработка ошибок инструмента)

Реализуйте функцию `get_weather(city: str)`, которая для неизвестного города возвращает не исключение, а структурированную ошибку `{"error": "city not found"}`. Напишите агента, который при получении такой ошибки пробует сначала проверить название города через поиск (имитация) и повторяет вызов. Покажите работу на примере с реальным или моковым API.

Демонстрирует: обработку ошибок, цикличность вызовов.

Задание 6 (МИП). Генерация игрового уровня через промпт

Разработайте системный промпт для LLM (например, YandexGPT), который генерирует описание игрового уровня для гоночной аркады: трасса (длина, количество поворотов, типы покрытия), расстановка бустов и препятствий, стартовая сетка. Затем напишите Python-скрипт, который отправляет этот промпт модели и выводит результат в формате JSON. Оцените, насколько сгенерированный дизайн пригоден для прямой реализации в Unity (укажите, что требует доработки).

Демонстрирует: промпт-инжиниринг, структурированный вывод, понимание игрового дизайна.

Задание 7 (МИП). Проектирование RL-среды для задачи преследования

Спроектируйте упрощённую среду для обучения с подкреплением (без реализации кода обучения). Опишите:

- Пространство наблюдений (не менее 4 параметров).
- Пространство действий (дискретное или непрерывное).
- Функцию награды (формула с пояснением).
- Условия завершения эпизода (столкновение, выезд за трассу, успешное удержание дистанции).
- Как вы будете оценивать, что агент действительно научился преследовать цель, а не просто следовать по траектории.

Демонстрирует: понимание RL, постановку задачи, умение формализовать награду.

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

Общие положения

Оценивание по дисциплине проводится в форме зачёта с дифференцированной оценкой («зачтено» / «не зачтено»). Для допуска к зачёту студент должен выполнить:

- не менее 80% лабораторных работ (9 из 11) с положительной оценкой;
- успешно пройти текущий контроль (тесты по лекциям).

Методические рекомендации, определяющие процедуры оценивания на зачете:

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 вопросов к зачету и результатов текущего контроля.

Зачет по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения зачета: устно.

Результат сдачи зачета заноситься преподавателем в зачетную ведомость и зачетную книжку.

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

Зачётная оценка формируется из двух составляющих:

Компонент	Доля в итоговой оценке	Форма проверки
Теоретическая часть (ответы на вопросы)	50%	Устный / письменный опрос по билетам (30 вопросов)
Практическая часть (решение кейсов / заданий)	50%	Выполнение практического задания (один из вариантов)

Процедура проведения зачёта

Зачёт проводится в два этапа:

Теоретический опрос – студент отвечает на 2 вопроса из билета (каждый билет содержит случайную выборку из 30 вопросов, приведённых в ФОС). Время подготовки – до 10 минут. Ответ – устно или письменно по решению преподавателя.

Практическое задание – студент получает один из 5–10 вариантов кейса (на основе лабораторных работ). Требуется написать код, объяснить логику или исправить ошибки. Время выполнения – 30 минут.

Критерии принятия решения:

- «Зачтено» – студент набрал не менее 60% от максимального балла (сумма за теорию и практику).
- «Не зачтено» – менее 60% или грубые ошибки в безопасности/архитектуре агента.

Критерии оценки:

1. Оценка ответов на вопросы к зачету (50% итоговой оценки)

Зачет

Полные, развернутые ответы с демонстрацией глубокого понимания темы.

Ответы содержат основные идеи, но без углубленного анализа.

Использование примеров, формул, корректных терминов.

Возможны небольшие ошибки в деталях или формулировках.

Умение анализировать и сравнивать методы.

% выполнения: 60–100% (допускаются незначительные неточности).

Незачет

Отсутствие понимания ключевых концепций.

Грубые ошибки или неспособность ответить на большую часть вопросов.

% выполнения: <60%.

2. Оценка выполнения практических кейсов и лабораторных работ (50% итоговой оценки)

Практическое задание на зачёте оценивается по шкале 0–50 баллов. Критерии:

Блок	Макс. балл	Показатели
Работоспособность кода/алгоритма	20	– Код запускается без ошибок (5)

Блок	Макс. балл	Показатели
		– Реализована требуемая функциональность (10) – Обработаны краевые случаи (5)
Корректность использования инструментов (API, Function Calling, RAG, MCP)	15	– Правильный выбор метода (5) – Корректное описание функций/инструментов (5) – Управление контекстом (previous_response_id, память) (5)
Стиль и читаемость (для кода) / Логика рассуждений (для текстового описания)	10	– Наличие комментариев, осмысленные имена переменных (5) – Отсутствие дублирования, структурированность (5)
Безопасность и обработка ошибок	5	– Проверка входных данных, обработка исключений (3) – Отсутствие жёстко зашитых секретов (2)

Шкала перевода баллов практики в итоговую долю (50%):

45–50 баллов → 50% (максимум)

35–44 балла → 40%

25–34 балла → 30%

15–24 балла → 20%

0–14 баллов → 0%

Итоговая оценка за зачёт = (баллы за теорию, макс 50) + (баллы за практику, макс 50).

Если сумма ≥ 60 → «зачтено».

Методические рекомендации, определяющие процедуры оценивания лабораторных работ:

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 зачете;

– 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.

Лабораторные работы оцениваются в процессе семестра (текущий контроль). Каждая работа оценивается по накопительной системе. Для допуска к зачёту необходимо получить **«зачтено» не менее чем за 9 из 11 лабораторных работ.**

1) Критерии оценивания лабораторной работы

Лабораторная работа считается выполненной, если студент:

1. Представил код в виде Jupyter Notebook или Python-скрипта – 30%
2. Продемонстрировал работающий результат (вывод, диалог, график и т.д.) – 30%
3. Ответил на контрольные вопросы (2–3 вопроса по работе) – 20%
4. Оформил отчёт с выводами и анализом полученных результатов – 20%

Общая оценка – «зачтено» при $\geq 60\%$ от указанных критериев.

2) Порядок защиты лабораторной работы

- Каждая лабораторная работа защищается в течение 2 недель после её выполнения.

- Защита включает:

- Демонстрацию кода и результатов.

- Устный ответ на вопросы преподавателя по теме работы (например: «Как бы вы изменили системный промпт для улучшения результата?», «Какие альтернативные способы реализовать RAG?»).

- При неудовлетворительной защите студент может пересдать работу после доработки (штраф 10% от баллов за повторную сдачу).

3) Пример индивидуального оценочного листа для лабораторной работы №3 «Агентный цикл генерации изображений»

Критерий	Макс. балл	Получено	Комментарий
Код запускается без ошибок (инициализация клиентов, функции)	5		
Реализован агентный цикл (генерация промпта → рисование → VLM → улучшение)	10		
Использовано структурное поле stop_condition для выхода из цикла	5		
Проведены эксперименты с разными концепциями (не менее 2)	5		
Оформлен вывод: помогает ли цикл улучшить качество?	5		
Ответы на контрольные вопросы	10		
Итого	40		Зачтено при ≥ 24

Виды нарушений и реакции на них

Нарушение	Результат
Код не запускается из-за синтаксических ошибок	Работа не принимается, возвращается на доработку

Использование готового чужого кода без ссылки (плагиат)	Оценка «не зачтено», до следующего семестра
Жёстко зашитый API-ключ в коде	Снижение на 10% от максимального балла работы (предупреждение)
Пропуск защиты без уважительной причины	Возможность защиты только в конце семестра со штрафом 20%

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

5.1 Основная литература:

1. Бессмертный, И. А. Искусственный интеллект. Введение в многоагентные системы : учебник для вузов / И. А. Бессмертный. — Москва : Издательство Юрайт, 2026. — 148 с. — (Высшее образование). — ISBN 978-5-534-20348-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589921>
2. Алфимцев А.Н. «Мультиагентное обучение с подкреплением». Издательство МГТУ им. Н.Э.Баумана, 2022
3. Хаген М. – Генеративные глубокие нейронные сети. Трансформеры и диффузионные модели (перевод) – Издательство АЛИСТ, 2026. <https://www.biblioglobus.ru/product/11084519?ysclid=movyvdzk2z98043917>
4. (МИП)Micheal Lanham. Learn Unity ML-Agents - Fundamentals of Unity Machine Learning: Incorporate new powerful ML algorithms such as Deep Reinforcement Learning for games. Packt Publishing Ltd, 2018

5.2 Дополнительная литература:

1. Ф. Джеймс, Т. Майк Промт-инжиниринг для GenAI. Паттерны надежных запросов для качественных результатов, Sprint Book, 2025, 432 с.
2. А.А. Костин Промпт-инжиниринг. Язык будущего. 2025. Ridero, 594 с.
3. В.А. Петров, А.Н. Тихонов. Искусственный интеллект и обработка естественного языка.
4. Златопольский Д. М. Основы программирования на языке Python. 2-е изд. Москва: ДМК Пресс, 2018.
5. А.А. Кузнецов. Машинное обучение для обработки естественного языка.
6. Руководство Google по промт-инжинирингу. Часть 1: основы и базовые техники <https://habr.com/ru/articles/901426/>
7. Документация облачных сервисов cloud.ru, yandex.cloud.
8. <https://education.yandex.ru/handbook/prompting>
9. <https://yandex.cloud/ru/training/training-pro#Data>
10. Elshall AS and Badir A (2025) Balancing AI-assisted learning and traditional assessment: the FACT assessment in environmental data science education. Front. Educ. 10:1596462. doi: 10.3389/feduc.2025.1596462 (Статья о рисках снижения базовых навыков при чрезмерной опоре на ИИ).
11. Wharton Knowledge “Without Guardrails, Generative AI Can Harm Education” (2024) <https://knowledge.wharton.upenn.edu/article/without-guardrails-generative-ai-can-harm-education/#:~:text=practice%20session%2C%20yet%20scored%20about,exam%20as%20the%20control%20group> Статья– эксперимент о влиянии GPT на обучение: зависимость от ИИ снижает глубокое усвоение материала.

12. (МИП) Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press. (главы 3–6, 13 – для понимания основ RL)
13. (МИП) Документация Unity ML-Agents: <https://github.com/Unity-Technologies/ml-agents>
14. (МИП) OpenAI Spinning Up in Deep RL (краткое введение) – <https://spinningup.openai.com/>
15. МИП: Примеры промптов для генерации игрового кода: «Prompt Engineering for Game Development» – онлайн-ресурс сообщества CarX Technologies (доступ по запросу).
16. Kadurin, Artur, et al. "The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology." *Oncotarget* 8.7 (2016): 10883.
17. Kadurin, Artur, et al. "druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico." *Molecular pharmaceutics* 14.9 (2017): 3098-3104.
18. Polykovskiy, Daniil, et al. "Molecular sets (MOSES): a benchmarking platform for molecular generation models." *Frontiers in pharmacology* 11 (2020): 565644.
19. Khrabrov, Kuzma, et al. " ∇^2 DFT: A Universal Quantum Chemistry Dataset of Drug-Like Molecules and a Benchmark for Neural Network Potentials." *Advances in Neural Information Processing Systems* 37 (2024): 36869-36889.
20. Polykovskiy, Daniil, et al. "Entangled conditional adversarial autoencoder for de novo drug discovery." *Molecular pharmaceutics* 15.10 (2018): 4398-4405.
21. Николенко, Сергей, Кадури, Артур и Архангельская Екатерина. Глубокое обучение. "Издательский дом" Питер", 2017.
22. <https://openreview.net/forum?id=FMMF1a9ifL>
23. <https://openreview.net/forum?id=ElUrNM9U8c#discussion>
24. <https://openreview.net/forum?id=JoO6mtCLHD>
25. <https://aclanthology.org/2024.findings-emnlp.760/>
26. <https://aclanthology.org/2020.coling-main.588/>
27. https://link.springer.com/chapter/10.1007/978-3-030-72113-8_30
28. https://link.springer.com/chapter/10.1007/978-3-031-42448-9_10
29. <https://aclanthology.org/2024.findings-naacl.288/>
30. Sun, X., Li, J., Kovalenko, A.V., Feng, W., Ou, Y. Integrating Reinforcement Learning and Learning From Demonstrations to Learn Nonprehensile Manipulation //IEEE Transactions on Automation Science and Engineering, 2023, 20(3), 1735–1744, DOI: 10.1109/TASE.2022.3185071, Q1
31. Petukhova, A.V.; Kovalenko, A.V.; Ovsyannikova, A.V. Algorithm for Optimization of Inverse Problem Modeling in Fuzzy Cognitive Maps. *Mathematics* 2022, 10, 3452. DOI: 10.3390/math10193452, Q1
32. Kirillova, E.; Kovalenko, A.; Urtenov, M. Study of the Current–Voltage Characteristics of Membrane Systems Using Neural Networks. *AppliedMath* 2025, 5, 10. <https://doi.org/10.3390/appliedmath5010010>,

5.3. Периодические издания:

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

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

Электронно-библиотечные системы (ЭБС):

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

Профессиональные базы данных

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ)) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature Protocols and Methods:
13. <https://experiments.springernature.com/sources/springer-protocols>
14. Springer Materials: <http://materials.springer.com/>
15. Nano Database: <https://nano.nature.com/>
16. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
17. "Лекториум ТВ" <http://www.lektorium.tv/>
18. 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
4. (МИП) Hugging Face Deep RL Course: <https://huggingface.co/learn/deep-rl-course/unit0/introduction>

Ресурсы свободного доступа

1. Промпт и Промпт инжиниринг https://courses.sberuniversity.ru/generative_art/img/13
2. Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4 (<https://arxiv.org/abs/2312.16171>).
3. Nathan Hunter. The Art of Prompt Engineering with ChatGPT
4. ChatGPT Prompt Engineering for Developers <https://www.promptingguide.ai/>
5. КиберЛенинка <http://cyberleninka.ru/>;
6. Американская патентная база данных <http://www.uspto.gov/patft/>
7. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
8. Федеральный портал "Российское образование" <http://www.edu.ru/>;
9. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
10. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .

11. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
12. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
13. Служба тематических толковых словарей <http://www.glossary.ru/>;
14. Словари и энциклопедии <http://dic.academic.ru/>;
15. Образовательный портал "Учеба" <http://www.ucheba.com/>;
16. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

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

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

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

По курсу предусмотрено проведение лекционных занятий, на которых дается систематизированный материал. В ходе лекций рассматриваются ключевые концепции.

Лабораторные занятия курса посвящены практическому освоению предложенных технологий и подходов.

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

Важнейшим компонентом курса является самостоятельная проектная работа, в ходе которой студент разрабатывает законченное решение для решения задач кейсов (пункт 4.1).

7. Материально-техническое обеспечение по дисциплине (модулю)

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

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
---	---------	--------------------	--------	---------------------	----------

1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб
		Аренда публичного IP	1		Шт
2	Виртуальная машина с GPU	Виртуальная машина с GPU NVIDIA® Tesla® V100 2 GPU 8 vCPU 128 Гб RAM	1	1	Шт
		ОС Ubuntu_24.04	1		Шт
		Системный диск SSD	1		Шт
			2000		Гб
		Диск SSD	1		Шт
			4096		Гб
		Диск SSD	1		Шт
			4096		Гб
		Аренда публичного IP	1		Шт
3	K8S	Master node 8 vCPU 16 RAM	1	1	Шт
		Worker node 10% доля 4 vCPU 32 RAM	5		Шт
		Worker node SSD-NVME	64		Гб
		Аренда публичного IP	1		Шт
4	ML Inference Instance Type GPU	Время работы в месяц	40	1	Ч
		Инстанс 8 x NVIDIA® H100 NVLink PCIe 160 vCPU 1520 GB RAM	1		Шт
		Количество запросов к ML-моделям	1		Млн. Шт
		Кэш ML-моделей	160		Гб
5	LLM	Токены GigaChat 2 Max	50		Млн. Шт
		Токены Embeddings	400		Млн. Шт

Дополнительные облачные ресурсы предоставляются технологическим партнером Yandex Cloud.

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебные аудитории для проведения занятий лекционного типа	Мебель: учебная мебель Технические средства обучения: экран, проектор, компьютер ауд. 129, 131, А-305, А-307	MS Office Word 2016 и выше Ms Power Point 2016 и выше
Учебные аудитории для проведения текущего контроля (Ауд. 101, 102, 105/1, 106 и 106а)	Мебель: учебная мебель Технические средства обучения: Экран, компьютер	Браузер Google Chrome, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim, NLTK, PyMorphy, фреймворком PyTorch)

	Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации	
Учебные аудитории для проведения промежуточной аттестации (Ауд. 129, 131, А-305, А-307)	Мебель: учебная мебель	-
Учебные аудитории для проведения лабораторных работ (Ауд. 101, 102, 105/1, 106 и 106а)	Мебель: учебная мебель Технические средства обучения: экран, компьютер Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации	Браузер Google Chrome, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim, NLTK, PyMorphy, фреймворком PyTorch)

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	MS Office Word 2016 и выше Ms Power Point 2016 и выше
Помещение для самостоятельной работы обучающихся (Ауд. 101, 102, 103, 105/1, 106 и 106а)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	Браузер Google Chrome, Matlab, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim, NLTK, PyMorphy, фреймворком PyTorch)

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта

между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.