

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ))



Рабочая программа дисциплины (модуля),
как компонент образовательной программы
специализированного высшего образования
по направлению подготовки
09.04.01 Информатика и вычислительная техника,
утвержденной первым проректором РУТ (МИИТ)
Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Генеративный искусственный интеллект

Направление подготовки: 09.04.01 Информатика и вычислительная техника

Направленность (профиль): Технологии проектирования программного обеспечения

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде
электронного документа выгружена из единой
корпоративной информационной системы управления
университетом и соответствует оригиналу

Простая электронная подпись, выданная РУТ (МИИТ)
ID подписи: 5665
Подписал: заведующий кафедрой Нутович Вероника
Евгеньевна
Дата: 01.09.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина направлена на формирование у магистров компетенций в области архитектурного проектирования программных систем с интеграцией компонентов генеративного искусственного интеллекта. В условиях острого кадрового дефицита AI-архитекторов курс готовит специалистов, способных не обучать нейросети с нуля, а грамотно встраивать большие языковые модели и мультиагентные системы в корпоративный ИТ-ландшафт. Студенты освоят парадигму AI-ассистированной разработки, методы промпт-инженерии, проектирование RAG-архитектур и оркестрацию автономных агентов с использованием отечественного стека и open-source решений. Практическая работа включает создание design-документов, расчет стоимости владения, настройку защитных барьеров и прототипирование enterprise-систем. Выпускник сможет принимать обоснованные архитектурные решения, обеспечивая надежность, безопасность и экономическую целесообразность внедрения ИИ-сервисов в реальные бизнес-процессы.

Целью освоения дисциплины является формирование у обучающихся системных знаний и практических умений в области архитектурного проектирования, интеграции и безопасной эксплуатации программных систем, использующих технологии генеративного искусственного интеллекта и мультиагентной оркестрации в корпоративной среде.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности: анализировать предметную область и выбирать оптимальные ИИ-компоненты для решения бизнес-задач, проектировать масштабируемые RAG-архитектуры и мультиагентные топологии с применением отечественных LLM API, разрабатывать стратегии промпт-инженерии и защитные барьеры для минимизации рисков галлюцинаций и утечек данных, рассчитывать совокупную стоимость владения ИИ-инфраструктурой и оформлять техническую документацию в соответствии с требованиями стандартов предприятия.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ПК-5 - Способен проектировать и разрабатывать программные продукты с применением перспективных методов исследования на основе

мировых тенденций развития вычислительной техники и информационных технологий.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- концепция генеративного искусственного интеллекта как инструмента программной инженерии и архитектурного компонента современных информационных систем;

- эволюция больших языковых моделей и их интерфейсов прикладного программирования от базовых completion API до современных агентов с инструментальным взаимодействием;

- архитектура нейросетей типа трансформер на концептуальном уровне, механизмы внимания и контекстного обучения, необходимые для понимания ограничений LLM;

- принципы работы эмбединг-моделей и векторных представлений данных для задач семантического поиска и RAG-систем;

- основы prompt engineering как инженерной дисциплины, включая техники Zero-shot, Few-shot, Chain-of-Thought, ReAct и Tree-of-Thought;

- методологии промпт-дизайна для сложных задач, включая структурирование системных промптов, управление контекстом и версионирование промпт-библиотек;

- архитектурные паттерны интеграции LLM как сервиса в распределенные программные системы, включая синхронные и асинхронные подходы;

- технологии и протоколы взаимодействия с отечественными и открытыми LLM API, управление токенами и стоимостью вызовов;

- архитектурные паттерны систем Retrieval-Augmented Generation (RAG), включая стратегии чанкинга, индексации и реранжирования документов;

- технологии векторных баз данных и их интеграция с LLM-сервисами в enterprise-архитектурах;

- концепция AI-агентов и их архитектура, включающая планирование, память, инструментальное взаимодействие и рефлекссию;

- фреймворки мультиагентной оркестрации и паттерны координации между агентами;

- технология Function Calling и инструментального взаимодействия агентов с внешними API, базами данных и корпоративными системами;

- методы оценки качества работы LLM и агентных систем, включая LLM-as-a-Judge, бенчмарки и пользовательские метрики;
- классификация рисков и уязвимостей интеграции генеративного ИИ, включая промпт-инъекции, утечки данных и галлюцинации;
- архитектурные паттерны AI Guardrails и защитных барьеров для обеспечения безопасности LLM-компонентов;
- методологии AI-ассистированной разработки программного обеспечения с использованием современных IDE-интеграций и отечественных аналогов;
- принципы верификации и валидации артефактов, сгенерированных ИИ (код, документация, архитектурные решения);
- технологии контейнеризации и развертывания ИИ-интегрированных сервисов с использованием платформ оркестрации;
- методы observability и мониторинга для LLM-сервисов и анализа расхода токенов;
- методы расчета стоимости владения (ТСО) программных систем с ИИ-компонентами и экономического обоснования архитектурных решений;
- стандарты ЕСПД и ГОСТ для оформления документации на системы с ИИ-компонентами, включая описание промпт-стратегий и архитектурных решений.

Уметь:

- проектировать архитектуру программных систем с интеграцией генеративных ИИ-компонентов при помощи нотаций C4 Model и паттернов AI-интеграции при условии соблюдения требований к надежности и стоимости в enterprise-среде;
- разрабатывать, тестировать и версионировать промпт-стратегии для получения предсказуемых результатов от LLM при помощи техник Chain-of-Thought и ReAct при условии необходимости обеспечения воспроизводимости генерации в production-системе;
- интегрировать отечественные LLM API в backend-сервисы программных систем при помощи специализированных SDK и асинхронных паттернов при условии обработки ошибок и контроля стоимости token-based pricing;
- проектировать и реализовывать RAG-архитектуры для работы с закрытыми корпоративными данными при помощи векторных СУБД и стратегий чанкинга при условии минимизации галлюцинаций и обеспечения контекстной точности;

- разрабатывать мультиагентные системы с оркестрацией AI-агентов при помощи открытых фреймворков и паттернов tool-use при условии необходимости автоматизации многошаговых задач;

- применять AI-ассистированные инструменты в цикле разработки ПО при помощи IDE-интеграций и кастомных промпт-стратегий при условии обязательной верификации сгенерированного кода;

- выявлять и митигировать риски интеграции генеративного ИИ при помощи AI Guardrails и архитектурных паттернов безопасности при условии соблюдения требований информационной безопасности;

- оценивать стоимость владения и производительность ИИ-компонентов при помощи методов финансового моделирования и инструментария observability при условии обоснования экономической целесообразности решений;

- проводить критическую верификацию артефактов, сгенерированных ИИ при помощи методов code review и статического анализа при условии обеспечения качества и соответствия корпоративным стандартам;

- оформлять техническую документацию на системы с ИИ-компонентами по стандартам ЕСПД при помощи отечественных офисных пакетов и AI-ассистентов при условии обязательной верификации сгенерированного содержимого.

Владеть:

- методами архитектурного проектирования и прототипирования enterprise-систем с ИИ-компонентами в условиях ограничений на вычислительные ресурсы и безопасность;

- инструментарием промпт-инженерии и фреймворками мультиагентной оркестрации для автоматизации сложных бизнес-процессов;

- навыками критической верификации артефактов, сгенерированных искусственным интеллектом, и настройки сред AI-ассистированной разработки;

- методиками расчета стоимости владения ИИ-сервисами и оформления проектной документации по стандартам ЕСКД.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 6 з.е. (216 академических часа(ов)).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов	
	Всего	Семестр №1
Контактная работа при проведении учебных занятий (всего):	64	64
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	32	32

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 152 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Генеративный ИИ как инструмент программной инженерии Рассматриваемые вопросы: - эволюция роли ИИ в цикле разработки ПО от автодополнения до автономных агентов; - классификация генеративных моделей по модальностям и применимости в программной инженерии; - концепция LLM как архитектурного компонента современных информационных систем; - обзор отечественного и международного рынка AI-инструментов для разработчиков.
2	Архитектура трансформеров на концептуальном уровне Рассматриваемые вопросы: - принцип работы механизма внимания и его влияние на возможности LLM; - контекстное окно и его ограничения при проектировании систем; - механизмы контекстного обучения (in-context learning); - современные семейства моделей и их специализация.

№ п/п	Тематика лекционных занятий / краткое содержание
3	Эмбединги и векторные представления данных Рассматриваемые вопросы: <ul style="list-style-type: none"> - математическая концепция плотных векторных представлений; - алгоритмы обучения эмбединг-моделей и метрики семантической близости; - применение эмбедингов для семантического поиска и кластеризации; - выбор эмбединг-модели под конкретную задачу и язык.
4	Основы Prompt Engineering как инженерной дисциплины Рассматриваемые вопросы: <ul style="list-style-type: none"> - анатомия эффективного промпта и структурирование запросов; - техники Zero-shot и Few-shot Learning для различных задач; - управление ролями, контекстом и форматом ответа; - версионирование и менеджмент промпт-библиотек в проектах.
5	Продвинутые техники Prompt Engineering Рассматриваемые вопросы: <ul style="list-style-type: none"> - Chain-of-Thought и Self-Consistency для сложных рассуждений; - методология ReAct (Reasoning + Acting) для агентных систем; - Tree-of-Thought и Graph-of-Thought для многовариантного анализа; - автоматическая оптимизация промптов и специализированные фреймворки.
6	Архитектурные паттерны интеграции LLM как сервиса Рассматриваемые вопросы: <ul style="list-style-type: none"> - синхронные и асинхронные паттерны взаимодействия с LLM API; - обработка ошибок, retry-стратегии и fallback-механизмы; - rate limiting и управление очередями запросов; - проектирование LLM Gateway и маршрутизация между моделями.
7	Интеграция с отечественными LLM API Рассматриваемые вопросы: <ul style="list-style-type: none"> - архитектура и возможности API ведущих отечественных провайдеров; - управление токенами, биллинг и оптимизация стоимости вызовов; - особенности работы с русскоязычными моделями; - унифицированные интерфейсы и миграция между провайдерами.
8	Архитектура RAG-систем Рассматриваемые вопросы: <ul style="list-style-type: none"> - компоненты RAG-пайплайна – индексация, retrieval, генерация; - стратегии чанкинга документов и их влияние на качество; - гибридный поиск и алгоритмы реранжирования; - оценка качества RAG и борьба с галлюцинациями.
9	Векторные базы данных для RAG-архитектур Рассматриваемые вопросы: <ul style="list-style-type: none"> - обзор современных векторных СУБД и расширений для реляционных БД; - архитектурные различия и сценарии применения; - индексация и оптимизация производительности; - интеграция с корпоративной инфраструктурой.
10	Концепция и архитектура AI-агентов Рассматриваемые вопросы: <ul style="list-style-type: none"> - компоненты агента – планировщик, память, инструменты, рефлексия; - классификация агентов по уровню автономности; - паттерны проектирования агентных систем; - жизненный цикл и state management агентов.
11	Мультиагентная оркестрация Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> - современные фреймворки мультиагентной оркестрации; - паттерны координации – иерархия, дебаты, конвейер; - распределение ролей и обмен сообщениями между агентами; - отладка и observability мультиагентных систем.
12	Tool-use и Function Calling в агентных системах Рассматриваемые вопросы: <ul style="list-style-type: none"> - технология Function Calling и ее реализация в различных LLM; - проектирование интерфейсов инструментов для агентов; - интеграция с внешними API, БД и корпоративными системами; - безопасность и валидация вызовов инструментов.
13	AI-ассистированная разработка ПО Рассматриваемые вопросы: <ul style="list-style-type: none"> - обзор инструментов AI-ассистированной разработки и отечественных аналогов; - методологии AI-Human Pair Programming; - кастомные промпт-стратегии для доменной разработки; - влияние на продуктивность и качество кода.
14	Безопасность и риски интеграции генеративного ИИ Рассматриваемые вопросы: <ul style="list-style-type: none"> - классификация атак – промпт-инъекции, jailbreaking, data extraction; - утечки конфиденциальных данных через LLM API; - галлюцинации и их последствия в production-системах; - правовые аспекты и compliance.
15	AI Guardrails и защитные архитектурные паттерны Рассматриваемые вопросы: <ul style="list-style-type: none"> - входная и выходная фильтрация промптов и ответов; - фреймворки построения защитных барьеров; - детекция токсичности и смещений (bias); - аудит-логирование и мониторинг аномалий.
16	Observability, TCO и документация систем с ИИ Рассматриваемые вопросы: <ul style="list-style-type: none"> - инструментарий мониторинга LLM-сервисов; - методы расчета стоимости владения и оптимизации token usage; - стандарты ЕСПД и ГОСТ для документации ИИ-систем; - верификация AI-сгенерированной документации и кода.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Анализ предметной области и выбор ИИ-компонентов для курсового проекта Студент анализирует требования к выбранной предметной области и формирует техническое задание на разработку программной системы с ИИ-компонентом. С использованием текстового редактора составляется аналитическая записка с обоснованием выбора между использованием внешнего API и локальной модели. Результатом работы является утвержденная концептуальная схема будущей системы и список критериев приемки.
2	Проектирование контекстного окна и потоков данных Студент проектирует схему передачи контекста между пользовательским интерфейсом и языковой моделью с учетом ограничений на длину входной последовательности. В среде моделирования

№ п/п	Тематика практических занятий/краткое содержание
	разрабатывается диаграмма последовательности, отражающая обработку длинных документов и стратегию их фрагментации. Итогом служит архитектурный паттерн управления контекстом для последующей программной реализации.
3	Выбор и обоснование эмбединг-моделей для семантического поиска Студент проводит сравнительный анализ доступных моделей векторизации текста по метрикам качества и производительности. С применением электронных таблиц формируется матрица принятия решений для выбора конкретной эмбединг-модели под задачи разрабатываемой системы. Разрабатывается схема предобработки текста перед его переводом в векторное представление.
4	Разработка базовых промпт-стратегий и шаблонов Студент формулирует набор системных и пользовательских промптов для решения типовых задач в рамках выбранной предметной области. Создается первичная библиотека промптов с указанием ожидаемых форматов вывода и ролевых установок. Проводится ручное тестирование шаблонов на предмет их однозначности и устойчивости к вариациям входных данных.
5	Проектирование сложных цепочек рассуждений и агентных промптов Студент разрабатывает многошаговые промпт-сценарии с использованием техник цепочки рассуждений для решения нетривиальных бизнес-задач. Составляется блок-схема логических переходов и точек ветвления внутри диалога с языковой моделью. Результат оформляется в виде структурированного документа, содержащего инструкции для алгоритмов рассуждения.
6	Архитектурное проектирование шлюза для LLM API Студент проектирует микросервис-посредник для маршрутизации запросов, обработки ошибок и реализации повторных попыток. С использованием нотаций UML разрабатываются диаграммы классов и компонентов, описывающие взаимодействие backend-сервиса с внешними провайдерами. Формируется спецификация асинхронных очередей для сглаживания пиковых нагрузок
7	Расчет стоимости владения и оптимизация токенов Студент проводит финансовое моделирование эксплуатации системы на основе тарифов отечественных провайдеров. В электронных таблицах рассчитывается прогнозная стоимость обработки различных сценариев использования и определяются точки оптимизации расхода токенов. Разрабатывается стратегия кэширования повторяющихся запросов для снижения операционных затрат.
8	Проектирование конвейера RAG и стратегий чанкинга Студент разрабатывает алгоритмическую схему конвейера извлечения и генерации с учетом специфики корпоративных документов. Определяются оптимальные параметры разбиения текста на смысловые фрагменты и стратегии их перекрытия. Результатом является детальное описание пайплайна подготовки данных и логики формирования контекстного промпта.
9	Проектирование схемы векторной базы данных Студент проектирует логическую и физическую структуру хранения векторных представлений и метаданных в реляционной базе данных. Разрабатывается схема индексов для обеспечения высокой скорости семантического поиска и формулируются запросы для гибридной фильтрации. Итогом служит скрипт инициализации базы данных и описание политик управления пространством.
10	Архитектура состояния и памяти AI-агента Студент проектирует механизм управления долгосрочной и краткосрочной памятью для автономного программного агента. Разрабатывается структура графа состояний, описывающая жизненный цикл выполнения многошаговой задачи и точки сохранения контекста. Формируются требования к интерфейсам чтения и записи в хранилище состояний.
11	Проектирование мультиагентной топологии и ролевой модели Студент разрабатывает схему взаимодействия нескольких специализированных агентов для решения комплексной задачи. Определяются роли, зоны ответственности и протоколы обмена сообщениями между компонентами системы. Результат оформляется в виде архитектурной диаграммы оркестрации с описанием механизмов разрешения конфликтов.

№ п/п	Тематика практических занятий/краткое содержание
12	Спецификация инструментов и API-контрактов для агентов Студент проектирует интерфейсы внешних инструментов, которые будут доступны агентам через механизм вызова функций. Разрабатываются схемы описания функций, спецификации входных параметров и форматов возвращаемых ошибок. Формируется реестр корпоративных API, к которым агенты получают санкционированный доступ.
13	Настройка среды AI-ассистированной разработки Студент анализирует и описывает конфигурацию плагинов для среды разработки, подключаемых к корпоративному шлюзу. Разрабатывается набор кастомных инструкций и правил для генерации кода в соответствии со стандартами предприятия. Результатом является документ с настройками среды и примерами верификации сгенерированных артефактов.
14	Анализ уязвимостей и моделирование угроз Студент проводит аудит спроектированной архитектуры на предмет уязвимостей к инъекциям и утечкам конфиденциальных данных. Составляется матрица рисков с оценкой вероятности и потенциального ущерба от различных векторов атак. Разрабатывается план мероприятий по минимизации выявленных угроз на уровне кода и инфраструктуры.
15	Проектирование защитных барьеров и фильтров Студент проектирует архитектуру слоя защитных барьеров для валидации входных запросов и фильтрации сгенерированных ответов. Разрабатываются регулярные выражения и логические правила для детекции токсичного контента и выхода за рамки заданной доменной области. Формируется схема интеграции защитного слоя в общий конвейер обработки запросов.
16	Разработка плана мониторинга и финальной документации Студент проектирует систему метрик и дашбордов для наблюдения за качеством работы ИИ-компонентов и расходом ресурсов. Определяются ключевые показатели эффективности и пороги срабатывания алертов. Формируется структура итоговой пояснительной записки курсового проекта в соответствии с требованиями стандартов.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Выполнение курсовой работы.
3	Подготовка к промежуточной аттестации.
4	Подготовка к текущему контролю.

4.4. Примерный перечень тем курсовых работ

1. Архитектурное проектирование корпоративной RAG-системы: стратегии интеграции векторного поиска и генеративных моделей для работы с технической документацией.
2. Разработка мультиагентной топологии: оркестрация автономных агентов для автоматизации процессов code review в CI/CD конвейере.
3. Проектирование отказоустойчивого API-шлюза: маршрутизация запросов между отечественными LLM-провайдерами с учетом метрик стоимости и задержек.

4. Архитектура защитного слоя (AI Guardrails): паттерны предотвращения промпт-инъекций и фильтрации токсичного контента в публичных ИИ-сервисах.

5. Проектирование интеллектуальной системы обработки документов: микросервисная архитектура с использованием мультимодальных языковых моделей.

6. Разработка корпоративного шлюза для AI-ассистентов: обеспечение изоляции конфиденциального кода при интеграции со средами разработки.

7. Архитектура гибридной системы GraphRAG: совмещение графов знаний и векторного поиска для сложной бизнес-аналитики.

8. Прототипирование изолированного инференс-кластера: развертывание открытых языковых моделей в защищенном корпоративном контуре.

9. Проектирование мультиагентной системы поддержки: автоматизация многошаговых сценариев решения инцидентов с использованием инструментального взаимодействия.

10. Архитектура подсистемы observability: финансовый и технический мониторинг потребления ресурсов в распределенных генеративных сервисах.

11. Разработка агентной системы тестирования: генерация и валидация интеграционных тестов на основе контрактных спецификаций.

12. Проектирование микросервиса управления промптами: версионирование, A/B-тестирование и оценка эффективности шаблонных запросов.

13. Архитектура интерфейса Text-to-SQL: обеспечение безопасного доступа бизнес-аналитиков к корпоративным хранилищам данных через генеративные модели.

14. Разработка паттернов graceful degradation: обеспечение отказоустойчивости систем, критически зависящих от внешних стохастических API.

15. Проектирование подсистемы аудита: логирование и трассировка генеративных артефактов для обеспечения соответствия требованиям регуляторов.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Душкин, Р. В. Генеративный искусственный интеллект : руководство / Р. В. Душкин. —	ЭБС Лань : [сайт]. — URL: https://e.lanbook.com/book/514902 (дата обращения: 10.06.2026)

	Москва : ДМК Пресс, 2025. — 228 с. — Текст : электронный	
2	Лекун, Я. Как учится машина: Революция в области нейронных сетей и глубокого обучения / Я. Лекун. — Москва : Альпина Паблишер. — Текст : электронный	ЭБС Лань : [сайт]. — URL: https://e.lanbook.com/book/213980 (дата обращения: 10.06.2026)
3	Глубокое обучение с TensorFlow, Keras и PyTorch. — Текст : электронный	ЭБС Лань : [сайт]. — URL: https://e.lanbook.com/book/514946 (дата обращения: 10.06.2026)
4	Паттерсон, Дж. Глубокое обучение с точки зрения практика / Дж. Паттерсон, А. Гибсон. — Текст : электронный	ЭБС Лань : [сайт]. — URL: https://e.lanbook.com/book/116122 (дата обращения: 10.06.2026)

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

официальные документации отечественных провайдеров LLM API и платформы для разработчиков;

руководства по промпт-инженерии и архитектуре мультиагентных систем;

справочные системы по стандартизации (ЕСКД, ГОСТ) и нормативно-правовому регулированию ИИ в РФ.

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Офисные пакеты: российские решения для работы с документами по ГОСТ;

Среда разработки: современные IDE с поддержкой AI-ассистентов и открытые редакторы кода;

Технологический стек ИИ: языки программирования, фреймворки оркестрации агентов и библиотеки векторного поиска;

СУБД и хранилища: реляционные базы данных с расширениями для векторного поиска и специализированные in-memory хранилища;

Инструменты observability: платформы для трассировки промптов и мониторинга стоимости инференса.

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

Для практических занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Курсовая работа в 1 семестре.

Экзамен в 1 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами»

И.С. Разживайкин

Согласовано:

Заведующий кафедрой ЦТУТП

В.Е. Нутович

Председатель учебно-методической
комиссии

Н.А. Андриянова